

UNIVERSIDAD CARLOS III DE MADRID



**Aplicación web para la evaluación semiautomática
de ejercicios de programación**

**ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA TÉCNICA DE TELECOMUNICACIONES:
TELEMÁTICA
PROYECTO FIN DE CARRERA**

Autor: Wafaa Agayou
Tutor: Luis de la fuente Valentín
Fecha: 12 de Julio 2012

INDICE GENERAL

1	INTRODUCCIÓN.....	9
1.1	RESUMEN	9
1.2	OBJETIVO	9
1.3	ESTADO DE ARTE	10
1.4	MOTIVACIÓN	10
1.5	GLOSARIO DE TÉRMINOS	11
2	ANÁLISIS Y DISEÑO	12
2.1	ANÁLISIS DE REQUISITOS FUNCIONALES	12
2.1.1	<i>Requisitos del profesor.....</i>	<i>12</i>
2.1.2	<i>Requisitos del alumno.....</i>	<i>13</i>
2.1.3	<i>Requisitos del Administrador</i>	<i>14</i>
2.2	HERRAMIENTAS DEL PROYECTO	15
2.3	REQUISITOS Y TECNOLOGÍAS	22
2.3.1	<i>Requisitos técnicos.....</i>	<i>22</i>
2.3.2	<i>Tecnologías</i>	<i>23</i>
2.4	DISEÑO TÉCNICO	26
2.4.1	<i>Aplicación.....</i>	<i>26</i>
2.4.2	<i>Base de datos.....</i>	<i>27</i>
2.5	PERFIL DE USUARIO	28
3	CASOS DE USO.....	30
3.1	ADMINISTRADOR.....	30
3.1.1	<i>Página principal</i>	<i>30</i>
3.1.2	<i>Activar el módulo</i>	<i>31</i>
3.1.3	<i>Desactivar el módulo</i>	<i>32</i>
3.1.4	<i>Desinstalar módulo.....</i>	<i>34</i>
3.1.5	<i>Modificar rol y datos del usuario.....</i>	<i>35</i>
3.1.6	<i>Eliminar usuario</i>	<i>37</i>
3.2	ALUMNO.....	38
3.2.1	<i>Página principal</i>	<i>38</i>
3.2.2	<i>Añadir Practica.....</i>	<i>40</i>
3.2.3	<i>Añadir versión</i>	<i>41</i>
3.2.4	<i>Histórico</i>	<i>42</i>
3.2.5	<i>Descargar test</i>	<i>43</i>
3.2.6	<i>Descargar enunciado</i>	<i>44</i>
3.2.7	<i>Descargar interfaz.....</i>	<i>44</i>
3.2.8	<i>Descargar mi entrega</i>	<i>44</i>
3.2.9	<i>Notificar la modificación del test.....</i>	<i>45</i>
3.3	PROFESOR	46
3.3.1	<i>Página principal</i>	<i>46</i>
3.3.2	<i>Añadir test</i>	<i>47</i>
3.3.3	<i>Descargar test</i>	<i>48</i>

3.3.4	<i>Descargar enunciado</i>	49
3.3.5	<i>Descargar interfaz</i>	50
3.3.6	<i>Practicas</i>	50
3.3.7	<i>Ver histórico</i>	51
3.3.8	<i>Modificar nota</i>	52
3.3.9	<i>Modificar test</i>	53
3.3.10	<i>Eliminar test</i>	54
3.3.11	<i>Ver todos los test</i>	54
3.3.12	<i>Modificar fecha fin</i>	55
4	IMPLEMENTACIÓN	56
4.1	CREACIÓN DEL MODULO	56
4.2	CREACIÓN DE LAS TABLAS EN LA BASE DE DATOS	62
5	INSTALACIÓN	65
5.1	INSTALACIÓN DE XAMPP	65
5.2	INSTALACIÓN DE DRUPAL	66
5.3	INSTALACIÓN DE IDIOMA	68
5.4	CONFIGURACIÓN DE DRUPAL	69
5.5	INSTALAR EL MÓDULO "DELIVER-AND-TEST"	72
6	CONCLUSIÓN Y TRABAJOS FUTUROS	75
6.1	CONCLUSIÓN	75
6.2	TRABAJOS FUTUROS	76
7	PRESUPUESTO	78
7.1	COSTES PERSONALES	78
7.1.1	<i>Investigación</i>	78
7.1.2	<i>Toma de requisitos</i>	78
7.1.3	<i>Diseño del sistema</i>	78
7.1.4	<i>Desarrollo de la aplicación web</i>	78
7.1.5	<i>Pruebas</i>	78
7.1.6	<i>Instalación de drupal</i>	79
7.1.7	<i>Desarrollo de la memoria</i>	79
7.2	COSTE TOTAL	79
7.2.1	<i>Costes materiales</i>	79
7.2.2	<i>Costes de recursos humanos</i>	79
8	BIBLIOGRAFÍA	81
	ANEXO I	84
	ANEXO II	84
	ANEXO III	84

INDICE DE FIGURAS

Figura 1: Estructura de instalación Drupal	24
Figura 2: El panel de control de XAMPP	25
Figura 3: Resumen del diagrama funcional del profesor	26
Figura 4: Resumen del diagrama funcional del alumno	27
Figura 5: diagrama de Flujo de Base de datos	28
Figura 6: pantalla principal de administrador	31
Figura 7: pantalla de activar module	32
Figura 8: pantalla de desactivar module	33
Figura 9: pantalla de desinstalar module	34
Figura 10: Pantalla de usuarios registrados	36
Figura 11: Pantalla de información de la cuenta del usuario registrado	37
Figura 12: Pantalla de eliminar usuario	38
Figura 13: Pantalla de página principal del alumno sin test modificado	39
Figura 14: Pantalla de página principal del alumno con test modificado	40
Figura 15: Pantalla del formulario añadir Practica	41
Figura 16 : Pantalla del formulario "Añadir versión"	42
Figura 17 : Pantalla de "Histórico"	43
Figura 18: pantalla de descargar un fichero	45
Figura 19: Pantalla de "Página principal"	46
Figura 20 : Pantalla del formulario "Añadir test"	48
Figura 21 : Pantalla de "Descargar test"	49
Figura 22 : Pantalla de "Practicas"	51
Figura 23: Pantalla de "Histórico" de las practicas	52
Figura 24: Pantalla del formulario "Modificar nota"	53
Figura 25: Pantalla del formulario "Modificar test"	54
Figura 26: Pantalla de "ver todos los test"	55
Figura 27: Pantalla del formulario "Modificar fecha fin"	55
Figura 28: Directorio del módulo "Deliver-and-test"	56
Figura 29 : Pantalla de usuarios	65
Figura 30: Pantalla de descargar Drupal 6.20	66
Figura 31: Pantalla de descargar Drupal 6.20	67
Figura 32: Pantalla de directorios de descarga	67
Figura 33: Pantalla de descargar la traducción español	68
Figura 34: pantalla de directorio de descarga	69
Figura 35: pantalla de acceso a panel de control XAMPP	69
Figura 36: pantalla de panel de control	70
Figura 37: Pantalla de phpAdmin	70
Figura 38: Pantalla de crear un usuario administrador	71
Figura 39: Directorio del module	72
Figura 40: pantalla de acceso a la aplicación web "deliver-and-test"	73
Figura 41: pantalla de autenticación	73
Figura 42: pantalla de activar módulo	74
Figura 43: Tabla con la duración de las tareas en semanas	79
Figura 44: Resumen de los gastos del desarrollo del proyecto	80

Agradecimientos

Quisiera agradecer a todas esas personas que me han apoyado tanto durante este tiempo. A mis dos madres (Naima y Aicha), mi abuela, mi padre, mis hermanas que siempre tienen palabras que te suben el ánimo y que te empujan a seguir adelante, a mi tutor Luis que ha estado pendiente todo el tiempo y presente siempre que he necesitado su ayuda y por supuesto a la persona más importante de mi vida que es mi marido y que ha estado siempre a mi lado, tanto en los momentos buenos como en los momentos más difíciles.

GRACIAS A TODOS

1 Introducción

1.1 Resumen

La presente memoria describe el proceso de análisis, diseño, desarrollo e implementación del proyecto “Aplicación web para la evaluación semiautomática de ejercicios de programación”. La aplicación denominada “Deliver-and-test” es una herramienta de evaluación y de entrega que usan los profesores y alumnos en asignaturas o cursos de programación. La herramienta está pensada para que sea de uso público y cualquier usuario o institución educativa puede hacer uso de ella sin suponer ningún coste.

1.2 Objetivo

El objetivo de este proyecto es crear una aplicación que pretende automatizar el sistema de entrega/corrección de prácticas, facilitar la evaluación al profesor, la entrega de prácticas al alumno y la comunicación entre ambos.

La aplicación deber ofrecer al profesor y el alumno unas funcionalidades básicas y necesarias para realizar su entrega/corrección, que mencionadas a continuación.

1. Funcionalidades del profesor:

- Subir un enunciado a la aplicación.
- Subir un test asociado al enunciado, que tiene que superar el alumno.
- Acceder a las prácticas entregadas por los alumnos.
- Tener un seguimiento de los esfuerzos de los alumnos.
- Puntuar las prácticas.
- Añadir comentario a las prácticas.
- Modificar un test que conlleva modificar todas las prácticas que han superado la version anterior del test y notificar el cambio a los alumnos autores de las prácticas.

2. Funcionalidades del alumno:

- Subir una práctica a la aplicación.
- Subir varias versiones de una práctica.
- Cumplir los requisitos del test y superarlo.
- Visualizar los enunciados y test subidos por el profesor.
- Visualizar las versiones de una práctica, subidas a la aplicación.
- Visualizar los errores de compilación y de ejecución.
- Mejorar su metodología de trabajo.
- Visualizar su nota y comentarios puestos por el profesor.

1.3 Estado de arte

Según los artículos [1], [2] [3] y [4], en cursos basados en la programación, la comunicación entre alumnos y profesores es un bien valioso y escaso. Desafortunadamente, los medios de calificación utilizados hoy en día siguen siendo limitados por varios motivos que se explican en esta sección.

Cada rol implicado tiene sus limitaciones. El profesor se encuentra ante una situación complicada a la hora de corregir y evaluar los códigos implementados que recibe, cuando tiene un número elevado de alumnos asignados y de distintos cursos porque invierte mucho tiempo en ello, en notificar la nota a sus alumnos, no puede dedicar el tiempo suficiente para cada uno de ellos ni obtener un seguimiento de los progresos realizados durante el curso. Y el alumno no puede visualizar sus fallos antes de ser evaluado, no siempre recibe la solución a sus dudas a tiempo y tarda en recibir su nota.

Para solucionar parte de este hándicap que puede existir en la evaluación, hay que acotar las necesidades de ambos. Los profesores necesitan información acerca de los problemas de los alumnos para lograr los objetivos del proyecto, información sobre los progresos de los alumnos, con el fin de mejorar el planteamiento del test o ayudarle a resolverla y los alumnos necesitan una respuesta casi inmediata cuando se encuentran con un obstáculo, necesitan un empujón y ayuda durante el desarrollo de su práctica.

El propósito de automatizar el medio de calificación es poder solucionar parte de los problemas encontrados hasta ahora, mejorar la comunicación entre el profesor y alumno. Hoy por hoy hay varias aplicaciones que cubren ciertas necesidades pero no están del todo completas o esta personalizadas como por ejemplo PSGE (the program submission and grading environment) [5], Marmoset [5]

“Deliver-and-test” es una aplicación cuyo objetivo es cumplir todos los requisitos para facilitar el proceso de evaluación y entrega de prácticas en cursos de programación.

1.4 Motivación

Las herramientas o métodos de corrección de prácticas y comunicación profesor-alumno utilizados actualmente en curso de programación no satisfacen todas las necesidades de ambos actores. El profesor sigue dedicando mucho tiempo en atender al alumno, compilar, ejecutar y evaluar práctica por práctica o crear su propio programa para ello, sabiendo que puede tener más de un curso que repartir de distintos niveles, sin olvidar las limitaciones que sufre el alumno utilizando herramientas que le impiden ciertas facilidades como

comunicar con el profesor, ver el resultado de la ejecución de su práctica al instante, visualizar observaciones escritas por el profesor, tener oportunidad de enviar/subir más de una versión de una práctica, etc.

Hoy por hoy existen diversas aplicaciones que ofrecen soluciones a partes de los problemas actuales como los mencionados en el apartado **(1.3)** del estado del arte, **PSGE y Marmoset**. La idea del proyecto es crear una aplicación **"*Deliver-and-test*"** de pruebas que compila, ejecuta y evalúa, de forma automática, las prácticas de los alumnos siguiendo un patrón definido por el profesor. Esta herramienta pretende ahorrar un tiempo valioso tanto del alumno como del profesor, mejorar la comunicación entre ambos y hacerla de forma casi instantánea.

1.5 Glosario de términos

En la memoria hay términos que se utilizan con frecuencia y que representan los elementos fundamentales de la aplicación desarrollada. A continuación se detalla cada uno de esos elementos:

- ✓ **Profesor:** Tutela el aprendizaje del alumno
- ✓ **Alumno:** El aprendiz del tutor
- ✓ **Test:** Es una batería de pruebas que crea el profesor para comprobar el desarrollo correcto de las practicas entregadas por los alumnos.
- ✓ **Interfaz:** Es la colección de métodos abstractos y propiedades, en ellos se especifican qué se debe hacer sin implementación. Con la interfaz se consigue que todos los alumnos tengan la misma estructura de la práctica.
- ✓ **Enunciado:** Es un documento que explica los requisitos que se necesitan para realizar la práctica y poder aprobar el **test**.
- ✓ **Practica:** Es El código implementado por los alumnos siguiendo la estructura definida en la Interfaz y cumpliendo los requisitos del enunciado para superar el test que ha elegido y quiere aprobar.

2 Análisis y diseño

En esta sección se detalla el proceso de análisis y diseño de la aplicación. El objetivo del análisis es obtener una especificación detallada del sistema que se quiere desarrollar. El análisis establece con precisión las necesidades y condiciones del sistema, es una fase esencial que se considera como punto de partida para la realización del diseño de la aplicación.

La fase de diseño es un proceso donde se reflejan y se traducen los requisitos del sistema en una representación del software. El diseño debe implementar todos los requisitos especificados en la fase de análisis y debe reunir todas las necesidades de los usuarios principales de la aplicación.

2.1 Análisis de requisitos funcionales

El análisis de requisitos funcionales refleja todas las funcionalidades identificadas y clasificadas que debe cumplir la aplicación. Estas funcionalidades describen cada uno de los procesos que debe llevar a cabo la aplicación y representan la base para poder desarrollarla.

Cada rol definido en la aplicación tiene ciertos permisos para acceder a las funcionalidades que le corresponde. En *“Deliver-and-test”* existen tres roles, administrador, profesor y alumno. A continuación se detallan los requisitos funcionales de cada uno de ellos.

2.1.1 Requisitos del profesor

Cuando el profesor quiere enunciar un test, publica el **test**, el **enunciado**, la **interfaz** y lo notifica a los alumnos para que puedan tener acceso y empezar a desarrollar su práctica. El enunciado explica el objetivo de la práctica que tiene que desarrollar el alumno, la interfaz le obliga a seguir una estructura determinada y el test le ayuda a comprobar que su desarrollo es válido y cumple todos los objetivos explicados en el enunciado. La aplicación *“Deliver-and-test”* permitirá al profesor realizar las siguientes funcionalidades para facilitarle la gestión de sus tareas de corrección de las prácticas durante el curso:

1. Subir un **test**, esta explicado con más detalle en el caso de uso **(3.3.2)** que evalúa las prácticas entregadas por los alumnos. Este **test** es el que indica al alumno si la práctica está aprobada o no, se le asigna un rango de fechas que definen el tiempo permitido a los alumnos para entregar sus soluciones, si vence la fecha, el **test** no aparecerá en el listado de test en la pantalla del alumno.
2. Subir un enunciado **(3.3.2)** que explica en detalle el objetivo del **test** y sus requisitos para superarlo.

3. Subir una interfaz **(3.3.2)** es la estructura que debe seguir el alumno para implementar su código.
4. Visualizar las prácticas de los alumnos **(3.3.6)** que han elegido el **test**. Se muestran los datos del alumno, la última versión entregada, el resultado de la ejecución, la nota y un comentario. Si la práctica ha sido evaluada por el profesor aparece la nota y el comentario con los datos que insertó el profesor.
5. Visualizar todas las versiones de una práctica entregadas por un alumno **(3.3.7)**. Todas las versiones entregadas se mostrarán con los datos mencionados en el punto anterior.
6. Modificar la fecha fin para ampliar el plazo de entrega **(3.3.12)** El profesor puede recuperar el **test** y lo activa para que el alumno tenga la oportunidad de entregar su práctica en el plazo ampliado.
7. Puntuar la práctica del alumno **(3.3.8)** Después de revisar la práctica entregada el profesor puntúa el alumno.
8. Modificar la nota de la práctica **(3.3.8)** El profesor puede rectificar su puntuación anterior.
9. Añadir comentario u observación **(3.3.8)** cuando se evalúa la práctica del alumno. Es un campo importante donde el profesor comenta su impresión sobre la práctica.
10. Modificar el **test** **(3.3.9)** y actualizar automáticamente la última versión de la práctica de cada alumno que ha seleccionado el **test**. Eso significa que el alumno verá los siguientes campos nota, **comentario** con sus valores por defecto que son '0' y "----" y el campo resultado o de la compilación/Ejecución.
11. Visualizar todos los **test** subidos **(3.3.11)** mostrando su estado actual (activo o no).
12. Eliminar el test **(3.3.10)**.

2.1.2 Requisitos del alumno

En muchas ocasiones el alumno tiene dificultad para comunicarse con el profesor y resolver dudas de forma rápida, invierte mucho tiempo en compilar y ejecutar su práctica de forma manual cada vez que hace un cambio y puede confundirse a la hora de realizar la entrega definitiva o entregarla mal. Todas estas complicaciones se resuelven con la aplicación **"Deliver-and-test"** porque ofrece funcionalidades que solucionan los problemas que sufren los alumnos de programación. A continuación se detallan esas funcionalidades desarrolladas:

1. Visualizar todos los **test** existentes y que estén activos **(3.2.1)**. Todos los **test** que tienen la fecha de finalización anterior a la fecha actual se mostrarán en la pantalla del alumno.

2. Subir la práctica que se quiere entregar **(3.2.2)**, eligiendo el **test** que se quiere superar.
3. Subir versiones de la práctica las veces que quiera. El alumno tiene la oportunidad de entregar varias versiones de su práctica y mejorarla.
4. Obtener el resultado de la ejecución en el tiempo real. El alumno obtendrá al instante el resultado de la ejecución o compilación si hay errores.
5. Visualizar los errores de compilación y de ejecución. Todos los errores de ejecución o de compilación se muestran en el campo resultado.
6. Visualizar nota y el comentario de cada versión de práctica entregada. Si la práctica no ha sido puntuada por el profesor se muestran los campos con los siguientes valores la nota a '0' y el comentario "---", si la práctica ha sido evaluada se muestran los valores insertados por el profesor.
7. Visualizar su histórico **(3.2.4)**. El alumno visualiza todas las versiones entregados de la práctica elegida, con todos los campos mencionados anteriormente.
8. Descargar el test del profesor para ver su contenido **(3.2.5)**.
9. Descargar el enunciado del test **(3.2.6)**.
10. Descargar la interfaz **(3.2.7)** que utiliza el alumno de estructura para la implementación de su código.
11. Descargar el código de la práctica **(3.2.8)**.

2.1.3 Requisitos del Administrador

El administrador es un usuario de **Drupal** (Plataforma web utilizada para el desarrollo e la aplicación "**deliver-and-test**"), dado de alta al crear la base de datos del gestor de contenidos, que no tiene funcionalidad propia en la aplicación "**Deliver-and-test**" porque no tiene ningún rol de los roles definidos en la aplicación, profesor o alumno. Es un rol muy importante porque es el que dispone de permisos para la instalación, control de usuarios y la configuración de la aplicación web. Sus funcionalidades son:

- Instalar el módulo "**deliver-and-test**" para poder acceder a la aplicación web desarrollada.
- Desinstalar el módulo.
- Eliminar módulo.
- Modificar rol, el rol de todos los usuarios registrados tiene que estar controlados por un usuario, en este caso el administrador tiene permiso para poder modificar el rol un usuario.
- Eliminar usuario.

2.2 Herramientas del proyecto

Antes de empezar el desarrollo del proyecto se han estudiado las herramientas que pueden cubrir las necesidades y requisitos del proyecto, se han hecho una comparativa entre ellas y se ha elegido la más adecuadas para la aplicación ***“Deliver-and-test”***.

Hay dos formas de desarrollar la aplicación, la primera es implementar desde cero la interfaz de usuario, las funcionalidades básicas y las funcionalidades que requiere el proyecto y la segunda mediante un Gestor de contenidos CMS.

El sistema de gestión de contenidos CMS [6] es un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM), gestionan un elevado número de contenidos y de usuarios. Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la web, cómo son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencias de los usuarios, la estructura de la web, etc.

“Deliver-and-test” es una aplicación relativamente compleja donde una parte de sus funcionalidades requeridas están proporcionadas por CMS como servicio básico, como por ejemplo la interfaz de usuario, el alta de usuario y la carga de contenido. Además de los servicios básicos, el gestor de contenidos ofrece escalabilidad, facilidad de mantenimiento y la posibilidad de agregar mejoras futuras. Esas ventajas que brinda un gestor de contenidos facilitarán el desarrollo de la aplicación, su instalación y su mantenimiento.

Hoy en día existen dos tipos de gestores de contenido que se diferencian según sus características y funcionalidades:

- **Genéricos:** Ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Sirven para soluciones de gestión de contenidos, para soluciones de comercio electrónico, blogs, portales, etc. Existen muchos gestores de contenidos genéricos como ***Drupal, Joomla, Plone, Alfresco, Apache lenya***...etc. A continuación se definirán los gestores de contenidos :

❖ **Drupal:** Es una aplicación de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios [9].

Drupal se caracteriza por:

- Fácil instalación, actualización y personalización.
- Interfaz de usuario intuitiva y fácil de utilizar.

- Cientos de plantillas de diseño para transformar el estilo del sitio.
- Miles de módulos disponibles para ampliar funcionalidades.
- Extenso centro de ayuda y soporte técnico en línea.
- Fácil incorporación de textos, imágenes, videos, música, etc.
- Múltiples roles de usuarios con diversos niveles de permisos.
- Habilitación de comentarios enlazados en las entradas.
- Soporte de **mod_rewrite** para **URL's** limpias y accesibles.
- Enlazamiento de páginas adicionales, blogs, etc.

Sin embargo este gestor de contenidos tiene algunas desventajas como:

- Curva de aprendizaje muy lenta
- Falta de evaluación entre los varios módulos desarrollados que cumplen el mismo objetivo

Actualmente hay más de 490.000 sitios que utilizan Drupal , tales como MTV (<http://www.mtv.co.uk/>), Casablanca (<http://www.whitehouse.gov/>), Gobierno francés (<http://www.gouvernement.fr/>) y la página de Rafa Nadal (<http://www.rafaelnadal.com/>).

❖ **Joomla**: Es una aplicación de código abierto programada mayoritariamente en PHP bajo una licencia GPL **[10]** .Este gestor de contenidos se caracteriza por:

- Creación de tantas páginas como se necesiten
- Definición de menús y submenús
- Administración de imágenes y ficheros
- Administración de usuarios para crear contenidos específicos o acceso sólo para usuarios registrados.
- Diseño basado en plantillas que se puede modificar para adaptarlas a las necesidades del usuario.
- Creación de módulos adicionales para poder colocar contenidos en determinadas partes de la web.
- Sindicación de noticias para poder publicar contenidos RSS automáticamente.
- Gestión de Banners (Espacio publicitario insertado en una página de Internet) que permitirán tener publicidad en la web.

Sin embargo **Joomla** tiene algunos inconvenientes:

- El código HTML generado no es semántico: no trae grandes beneficios, en cuanto a usabilidad, accesibilidad, posicionamiento en buscadores y a la hora de retocar o cambiar el código.
- Ausencia de acciones sólidas del Grupo de Desarrollo para hacer la aplicación más accesible. Dependencia excesiva del Javascript en su Panel de Administración
- Limitación en las opciones para personalizar rangos de usuarios y permisos

Actualmente hay más de 25.000 sitios que utilizan **Joomla** tales como Linux

<http://linux.com>), Música(<http://crispystock.com>), Farmacia (<http://www.farmaciaevedat.com/index.php>), Sloporte TV(<http://www.soportetv.es/>), etc.

❖ **Plone**: Es una aplicación de código abierto, basado en Zope **[11]** (un servidor de aplicaciones web de código abierto escrito en el lenguaje de programación Python **[12]**)y programado en Python. Plone se caracteriza:

- Producción muy rápida
- Gestión de contenido deslocalizado
- Edición de las páginas en tiempo real
- Enfoque centrado en el usuario
- Localización de la interfaz en modo nativo
- Plantillas con estándares de la industria
- Motor de búsqueda completo, indexación en tiempo real
- Modulable, evolutivo y fácilmente personalizable
- Motor de workflow integrado.

Sin embargo tiene como inconvenientes:

- Escasa documentación comparando con otros gestores de contenido.
- Escasa popularidad en España
- Escases de servidores de alojamiento (Hosting)

Existen más de 22000 sitios desarrollados en PLONE, entre los cuales están la NASA

(<http://nasascience.nasa.gov>), Nokia (<http://qt.nokia.com>),
Creaa(<http://www.creaa.es/>), Ayuntamiento de Ávila
(<http://www.avilaturismo.com/>).

❖ **Alfresco**: es un sistema de administración de contenidos libre, basado en estándares abiertos y de escala empresarial [13]. Alfresco se caracteriza por:

- Gestión de los permisos desde un único punto-
- Instalación únicamente en el servidor
- Acceso al repositorio a través de Servicios Web, REST, WebScripts, etc ...
- Conexión desde Portales Corporativos, ERPs, CRMs, etc .
- Arquitectura Escalable y tolerante a fallos.
- Creación de reglas a través de Wizards para transformar el contenido o actuar en función de cambios.
- Foros de discusión sobre documentos.
- Creación de Flujos de trabajo visualmente.
- Auditoría completa.
- Búsqueda similar a Google o a Yahoo.

Alfresco tiene algunas desventajas como:

- La web Content Management (WCM) de **Alfresco** contiene errores importantes que en parte evitar la presentación de contenidos
- La presentación de contenidos es bastante lenta y consume muchos recursos porque el contenido de los formularios web se guardan como datos XML que debe ser transformado en HTML sobre la marcha o en caché para su uso posterior
- Alfresco es una gran empresa de contenidos del sistema de gestión de sitios muy grandes y con multitud de funcionalidades. Puede que sea demasiado complejo para sitios más pequeños.

Existe hoy en día más de 500 sitios desarrollados con Alfresco como:

Easport (http://www.easportsworld.com/en_US),
CELER(<http://www.celersol.com/CELER/>),
Activision(<http://www.activision.com/index.html>),etc...

❖ **Apache lenya**: Es una aplicación de código abierto escrito en Java, manejando el formato de representación XML [14] Apache lenya se caracteriza por:

- Facilidad de editar contenidos
- Basado enteramente en XML y por lo tanto compatible con la mayoría de los estándares web.
- Independiente de la base de datos, no necesita de ninguna base de datos, todo lo que gestiona **Lenya** se basa en ficheros XML almacenados en el mismo servidor. Esto no significa que no pueda conectar a bases de datos a través de lenguajes tipo PHP (ASP, JSP, PHP, etc).

Apache Lenya tiene como desventajas:

- Lentitud del gestor de contenido en asimilar la terminología, por ejemplo “assets” se refiere a cualquier documento, imágenes, o cosa que se enlace en una página que no sea texto.
- Complejidad de programación básica como añadir un enlace a otra página.
- Escasez de servidores de alojamiento (hosting) que soporten Lenya.

- **Específicos**

- **Wikis:** Sitio Web dónde todos los usuarios pueden colaborar en los artículos, aportando información o rescribiéndola. También permite espacio para comentarios u opiniones.

❖ **MediaWiki:** es un software para wikis libre programado en el lenguaje PHP [15]. La wiki tiene las ventajas siguientes:

- El soporte de extensiones, que permite tener funciones que no vienen dentro de mediawiki o integrarlo con otros sistemas.
- El soporte de plantillas, denominadas “Skins”, para darle otra apariencia a la Wiki.

Tiene como inconveniente el que no es un sistema amigable para el usuario, para alguien no técnico puede resultar bastante complicado; pero detrás de él hay un sistema muy robusto y completo.

- **Foros:** sitio que permite ayudar y debatir en línea temas en los que están interesados.

❖ **phpBB:** es un sistema de foros gratuito basado programado en PHP y lanzado bajo la Licencia pública general de GNU [16] phpBB se caracteriza por:

- Posibilidad de usar HTML, código BB y "emoticones" en los mensajes.
- Cambio de aspecto del foro: Los usuarios podrán variar el aspecto a su gusto (activado por el administrador)
- Edición de los mensajes, respuestas, temas ...
- Los usuarios, por sí mismos, podrán hacer cambios en los nombres (activado por el administrador)
- Mensajes privados entre los usuarios.
- Opción de ser avisado vía e-mail cuando un mensaje es respondido.

Sin embargo tiene algunos inconvenientes como:

- Dificultad de manejo de la interfaz del administrador.
- Lentitud a la hora de instalar mods
- Editor poco amigable

❖ **MyBB:** es un sistema de foros gratuito basado programado en PHP y lanzado bajo la Licencia pública general de GNU. Se caracteriza por su gran funcionalidad frente a los anteriores sistemas de foro mencionados anteriormente [17].

MyBB se caracteriza por:

- Cantidad ilimitada de foros y subforos, usuarios. registrados, temas visuales y mensajes almacenados.
- Buscador integrado.
- Panel de control de usuario y administrador.
- Fácil instalación de plugins y modificaciones sin necesidad de manipular el código.
- Feeds RSS generados automáticamente para las suscripciones a todo el foro o a subforos concretos.
- Posibilidad de modificar las plantillas (HTML para anidar el contenido del foro) y los temas visuales, y de exportarlos e instalar nuevos.
- Posibilidad de tener varios idiomas, temas visuales y plantillas activos al mismo tiempo en el foro, para que los usuarios puedan elegir entre ellos.

Como inconvenientes tiene:

- El diseño y los temas son poco amigables. (también cambiaría esta expresión como no son llamativos o algo así)
- Escases de plugins y mod´s.
- Poca personalización del perfil.

Para elegir el gestor de contenidos que se adecúa a las necesidades del proyecto hay que basarse en los requisitos de los usuarios, que en este caso son el profesor y alumno, y valorar, según las características de cada uno de los Gestores de contenido, el que cubra la mayoría de los requisitos establecidos.

Los puntos decisivos para elegir un gestor de contenidos u otro son:

- Lenguaje de programación
- Servidor
- Instalación y alojamiento
- Sencillez de implementación
- Sencillez de configuración
- Uso de roles y workflow
- Comunidad
- Integración
- Seguridad y escalabilidad
- Documentación
- Mantenimiento

A continuación [Tabla 1] se recopila los puntos decisivos indicados anteriormente y se realiza una comparativa entre 4 gestores de contenido. La evaluación está elaborada en base a la experiencia adquirida en el mundo laboral, trabajando con los gestores de contenido Plone y Alfresco, la información conseguida realizando búsqueda en los siguientes webs [9][10][13][19] y de la memoria de PFC: **“Análisis, diseño e implementación de una aplicación web para gestión de información policial municipal basada en Plone”** de la alumna Fátima Sánchez [20].

Características	Plone	Joomla	Alfresco	Drupal
Lenguaje de programación	Python	Java	Java	PHP
Servidor	Zope	Tomcat	Tomcat	LAMP
Instalación y alojamiento	Bueno	Excelente	Bueno	Excelente
Sencillez de implementación	Bueno	Excelente	Bueno	Excelente
Sencillez de configuración	Bueno	Excelente	Bueno	Excelente
Uso de roles y workflow	Excelente	Bueno	Excelente	Excelente
Comunidad	Bueno	Bueno	Bueno	Excelente
Integración	Excelente	Bueno	Excelente	Excelente
Seguridad y escalabilidad	Excelente	Bueno	Bueno	Bueno
Documentación	Bueno		Bueno	Excelente
Mantenimiento	Bueno	Bueno	Bueno	Bueno

Tabla 1: Características de los gestores de contenidos

En definitiva, se elige *Drupal* porque ofrece una plataforma adecuada para desarrollar la aplicación y asegura la continuidad del mantenimiento y crecimiento futuro, independientemente del proveedor que desarrolla el sitio.

2.3 Requisitos y tecnologías

2.3.1 Requisitos técnicos

Como se ha decidido en el apartado anterior (2.2), para desarrollar el proyecto se hará uso de Drupal [3] que es un sistema de gestión de contenido modular, de código libre, con licencia GNU/GPL, muy configurable, fácil de instalar, soporta en cualquier sistema operativo, un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web, escrito en PHP [18], desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

Drupal es un sistema modular cuya gran parte de funcionalidad esta delegada en una serie de módulos específicos. Los módulos en Drupal son ficheros con extensión .module que contienen funciones escritas en PHP. Estas funciones actúan como **hooks[21]** (funciones gancho), que son llamadas por Drupal durante sus procesos habituales de gestión de contenido y construcción de páginas web. Por ejemplo, cada vez que un nodo es creado, visualizado, modificado o borrado, Drupal llama una de estos **hooks** pasándoles el contenido del nodo. De esta forma los módulos tienen la posibilidad de modificar y adaptar la información a visualizar en las páginas web antes de que se mande definitivamente al

navegador. Los **hooks** no sólo son llamados cuando se gestionan los nodos, también se llaman en muchas otras circunstancias. Por ejemplo, cuando se comprueba si el usuario tiene acceso a la información que está intentando consultar, cuando se empieza a crear o está a punto de terminarse una página web, cuando se están creando o modificando comentarios, cuando se están creando los menús, y así un largo etcétera. Todas estas llamadas que realiza Drupal a los módulos posibilita que el sistema sea altamente "maleable", logrando que se pueda modificar prácticamente el funcionamiento entero del sistema sin tener que alterar en ningún momento el núcleo (core) del mismo. De hecho, muchas de las características básicas de Drupal, como puede ser la posibilidad de organizar el contenido en categorías, son tratadas en la práctica como módulos.

Por defecto, el paquete de Drupal viene con una serie de módulos que forman parte de lo que se conoce como el núcleo (el core). Además de los módulos del núcleo, existen más de 2000 módulos contribuidos. Estos módulos son programados y mantenidos por la comunidad de usuarios de Drupal. Una forma de desarrollar el proyecto "**Deliver-and-test**" es usar los módulos ya existentes que satisfacen sus necesidades y configurarlos. Es una opción válida y practica pero con un gran inconveniente que es la dependencia.

Al usar los módulos existentes se exige al usuario, antes de poder hacer uso de la aplicación web, descargar la misma versión de Drupal y todos los módulos necesarios para el buen funcionamiento de éste. Otra forma de desarrollar el proyecto es crear un módulo nuevo de Drupal implementando todas las necesidades y los requisitos definidos en el análisis funcional independientemente del resto de los módulos. Esta opción es más laboriosa y compleja pero facilita al usuario el acceso a la aplicación web instalando solo el módulo desarrollado y permite en un futuro insertar mejoras por otros desarrolladores siempre bajo supervisión de la comunidad de desarrolladores Drupal. La opción que se ha considerado óptima y práctica para el desarrollo del proyecto es la segunda, crear un nuevo módulo con todas las funcionalidades definidas en los requisitos.

2.3.2 Tecnologías

➤ **Drupal**

Drupal [9] que es un sistema de gestión de contenido, muy configurable, fácil de instalar, soportado en cualquier sistema operativo, un sistema dinámico, se edita utilizando un entorno Web, escrito en PHP [22]. Desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

Drupal se instala sobre otras aplicaciones, Apache, MySQL y PHP, como se muestra en la figura **[Figura 1]**

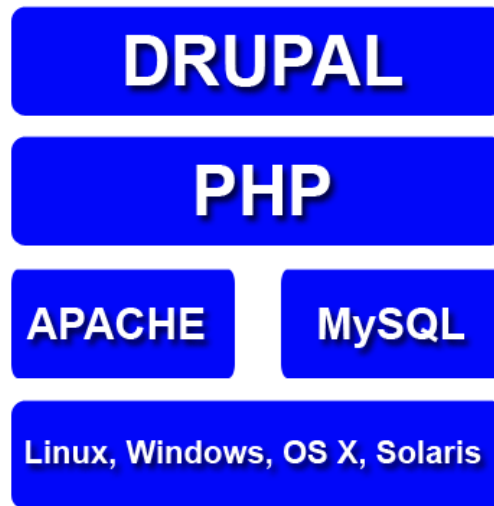


Figura 1: Estructura de instalación Drupal

➤ **MySQL**

MySQL es un sistema de gestión de bases de datos relacional con licenciado bajo la GPL de la GNU. Su diseño **multihilo** permite soportar una gran carga de forma eficiente.

Este gestor de bases de datos es el gestor más usado en el mundo del software libre, debido a su alto rendimiento, alta fiabilidad y facilidad de uso **[24]**.

➤ **Apache**

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web y para muchas otras tareas donde el contenido necesita ser puesto a disposición de una forma segura y confiable **[25]**.

➤ **PHP**

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools), es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+ [22].

En **“Deliver-and-test”**, se necesita instalar Drupal, Apache, SQL y PHP. en este proyecto se ha utilizado Xampp [26] un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. Está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X. [Figura 2]

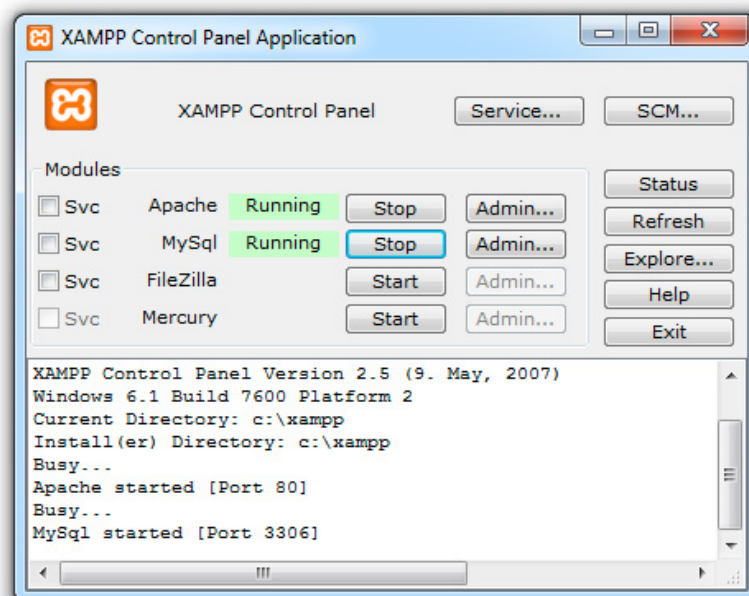


Figura 2: El panel de control de XAMPP

2.4 Diseño técnico

Un diagrama de flujo es una representación que ilustra la secuencia de las operaciones que se realizarán para conseguir la solución de un problema. En este proyecto los diagramas de flujo facilitan la comprensión del planteamiento realizado en el apartado 2.1 representándolo de una forma más sencilla y gráfica. Cada rol tiene su propio diagrama, menos el rol administrador ya que no tiene ninguna funcionalidad propia de “**Deliver-and-test**”. Se detalla a continuación los diagramas de flujo de alumno y profesor.

2.4.1 Aplicación

➤ Profesor

En el anexo se indica todo el funcionamiento de la aplicación de un usuario con rol de profesor. El documento en el anexo (ver anexo), muestra todas las operaciones paso a paso y su resultado desde que se autentica el profesor hasta que termine cualquier acción que quiere hacer.

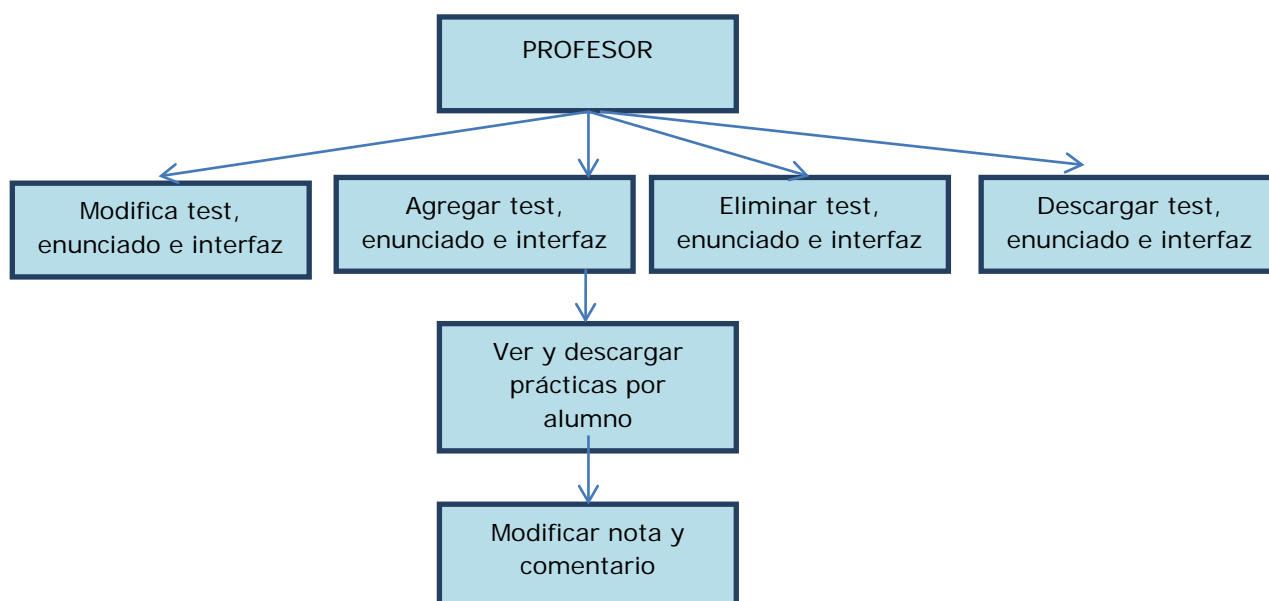


Figura 3: Resumen del diagrama funcional del profesor

➤ Alumno

En el anexo se indica todo el funcionamiento de la aplicación de un usuario con Rol Alumno. El documento anexo (ver anexos) muestra las operaciones que necesitan los alumnos para entregar sus prácticas y más funcionalidades que están detalladas en el documento.

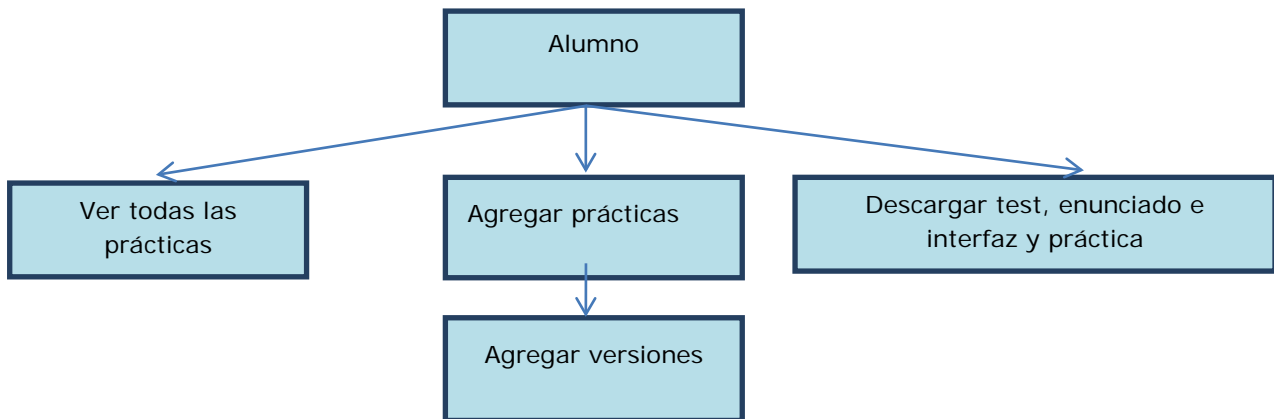


Figura 4: Resumen del diagrama funcional del alumno

2.4.2 Base de datos

Drupal crea en el proceso de su instalación todas sus propias tablas en la base de datos que se le asigna. Estas tablas almacenan todo el contenido insertado en la aplicación o la Web creada en el gestor documental. En este proyecto se ha decidido crear tablas que pertenecen al modelo de datos propio del módulo para utilizar en la menor medida, las tablas de *Drupal* y no crear ninguna dependencia. Las únicas tablas reutilizadas son la tabla **rol** y **user** donde se almacenan los roles que se crean al instalar el módulo y los usuarios que se registran en la aplicación Web.

Las tres tablas definidas en el modelo de datos del módulo, como se muestra en la *Figura 5* son:

- ❖ **Test:** almacena los datos del **test** subido por el profesor autenticado.
- ❖ **Practica:** almacena los datos de las prácticas subidas por el alumno autenticado.
- ❖ **test-profesor:** almacena la relación test-profesor creada al subir un test desde el formulario de **subir test** un profesor autenticado

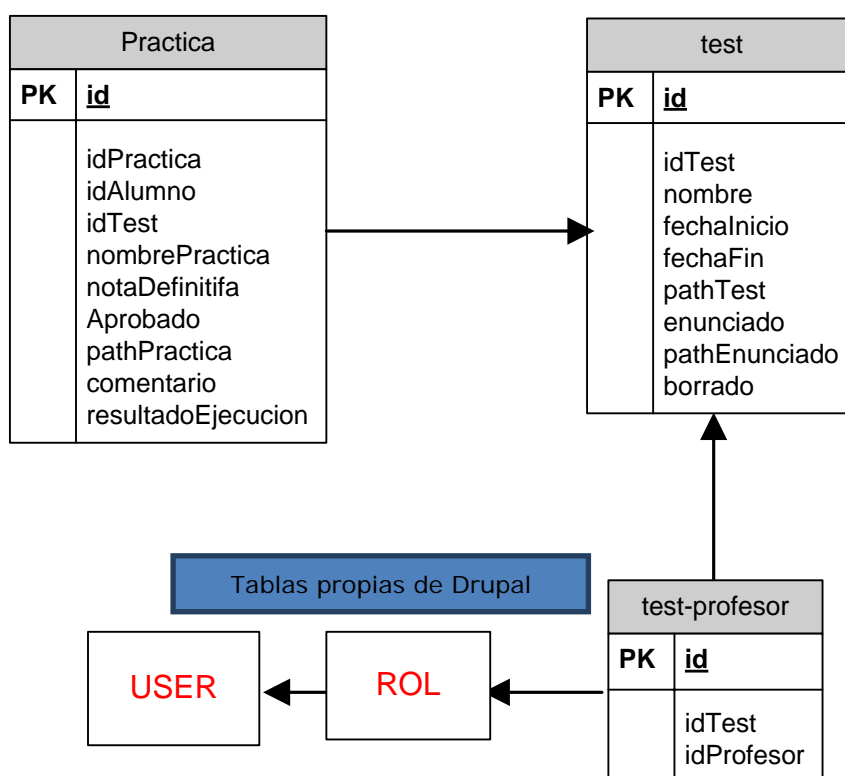


Figura 5: diagrama de Flujo de Base de datos

2.5 Perfil de usuario

Los usuarios de la aplicación web "**deliver-and-test**" tienen que tener un perfil establecido. En la aplicación existen dos perfiles, el profesor y el alumno que se crean al instalar el módulo. El usuario elige su rol cuando se registra por primera vez en la aplicación y según el rol elegido tiene acceso a las funcionalidades correspondientes, que se explican a continuación. Después de registrarse solo el administrador puede realizar cambios sobre el rol de los usuarios, sus nombres de usuario o sus contraseñas.

Una persona puede tener los dos perfiles existentes, profesor y alumno, siempre y cuando tenga dos usuarios distintos, si éste lo desea porque puede estar en las dos posiciones a la vez y necesite acceder a las funcionalidades de un rol u otro según la operación o la funcionalidad que quiere utilizar.

El perfil **profesor** permite al usuario acceder a las siguientes funcionalidades del profesor:

- ✓ Ver listado de test: muestra el listado de test del profesor autenticado subidos anteriormente.

- ✓ Ver prácticas de alumno: muestra las prácticas de los alumnos que han pasado el test elegido.
- ✓ Poner nota y comentario: formulario que permite al profesor puntuar al alumno y añadir un comentario sobre su práctica.
- ✓ Agregar test: formulario que permite al profesor agregar su test.
- ✓ Agregar enunciado: formulario que permite al profesor agregar el enunciado del test.
- ✓ Agregar interfaz: formulario que permite al profesor agregar la interfaz
- ✓ Modificar test, enunciado e interfaz: formulario que permite al profesor modificar su test, enunciado y/o interfaz.
- ✓ Modificar el resultado y la puntuación de las prácticas cuando se modifica el test que han elegido.
- ✓ Eliminar test: borra el test de forma lógica en la base de datos.

El perfil **alumno** permite al usuario acceder a las siguientes funcionalidades del alumno:

- ✓ Ver versiones de sus prácticas: muestra las versiones de la práctica seleccionada subida por el alumno
- ✓ Ver los **test** creados por el profesor: muestra todos los **test** subidos por todos los profesores que no estén finalizados
- ✓ Ver los enunciados: muestra los enunciados de los **test** mostrados
- ✓ Ver los interfaces: muestra los interfaces de los **test** mostrados
- ✓ Agregar practica: formulario que permite al alumno agregar una practica
- ✓ Agregar versión: Formulario que permite al alumno agregar una versión de la práctica seleccionada.
- ✓ Ver su nota y comentario: muestra la nota y comentario, modificados por el profesor, de la práctica seleccionada por el alumno.
- ✓ Compilar/Ejecutar la práctica subida y mostrar los resultados.

Los perfiles “profesor” y “alumno” se eliminan al desinstalar el módulo mientras que el rol de administrador se elimina cuando se desinstala o se borra la base de datos de Drupal.

3 Casos de uso

Un caso de uso es una descripción de los pasos o las actividades que se pueden realizar en la aplicación desarrollada. Cada caso de uso tendrá su descripción, los usuarios que puedan acceder en él, el resultado esperado después de ejecutar paso a paso el escenario que viene justo después del resultado y comentario para añadir observaciones adicionales.

Los tres perfiles definidos en la aplicación web tienen sus casos de uso que se explican a continuación.

3.1 Administrador

3.1.1 Página principal

La página principal es la primera página de Drupal que visualiza el administrador cuando se autentica con todas las configuraciones que puede manejar.

Definición	El administrador accede a su página principal después de autenticarse correctamente
Usuarios	Con rol de administrador
Resultado	Después de autenticarse correctamente el administrador visualiza la página de bienvenida de Drupal y el menú de configuración.
Escenario	Autenticarse con usuario con rol administrador Visualizar el menú en el lado izquierdo de la aplicación que contiene todas las opciones de configuración del sitio.

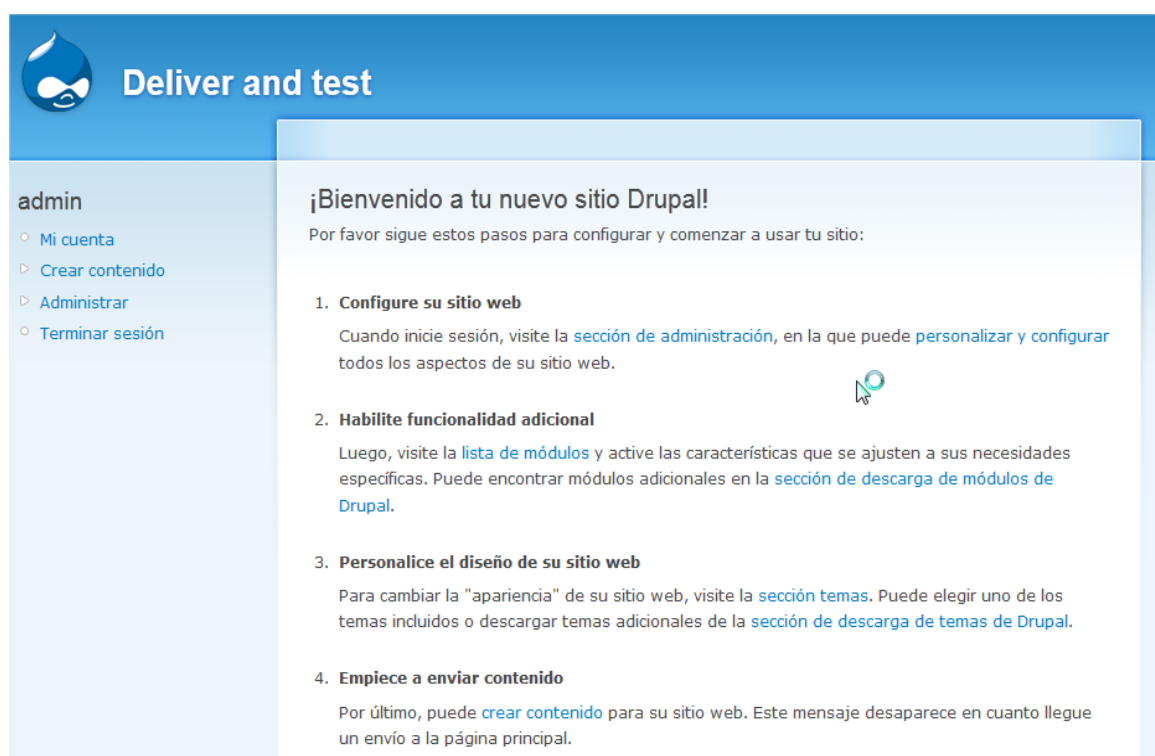


Figura 6: pantalla principal de administrador

3.1.2 Activar el módulo

Se explican los pasos para activar el módulo de la aplicación y poder acceder a sus servicios

Definición	El administrador activa el módulo <i>"deliver-and-test"</i> después de autenticarse correctamente
Usuarios	Con rol de administrador
Resultado	Después de autenticarse correctamente el administrador visualiza la página de bienvenida de drupal y el menú de configuración. Accediendo a la parte de módulo activa el <i>"deliver-and-test"</i>
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol administrador 2. Visualizar el menú en el lado Izquierdo de la aplicación que contiene todas las opciones de configuración del sitio. 3. Ir a Administrar→Configuración de sitio →Módulos. 4. Seleccionar el módulo <i>"deliver-and-test"</i>. 5. Pulsar el botón "guardar"

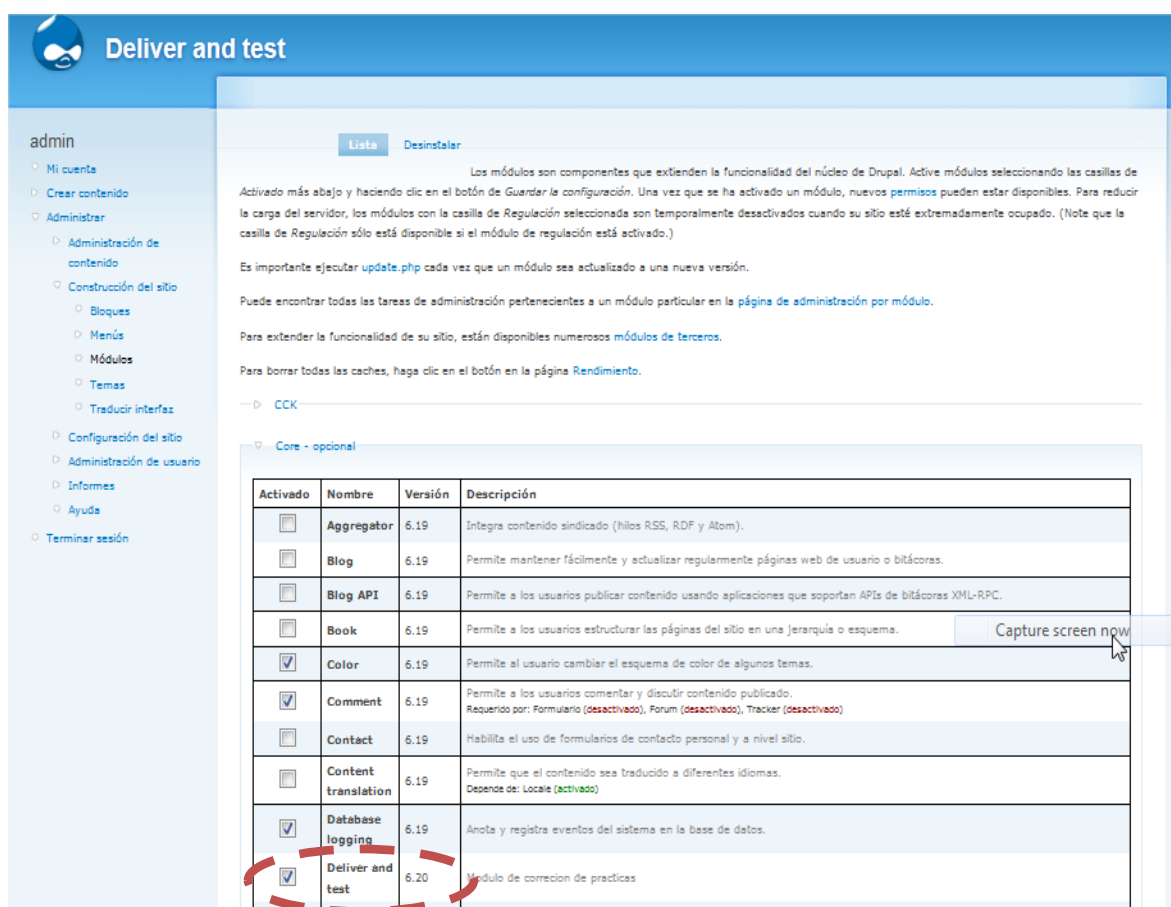


Figura 7: pantalla de activar module

3.1.3 Desactivar el módulo

Se desactiva el módulo de Drupal.

Definición	El administrador desactiva el módulo <i>“deliver-and-test”</i> después de autenticarse correctamente
Usuarios	Con rol de administrador
Resultado	Después de autenticarse correctamente el administrador visualiza la página de bienvenida de drupal y el menú de configuración. Accediendo a la parte de módulo desactiva el <i>“deliver-and-test”</i>

Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol administrador 2. Visualizar el menú en el lado Izquierdo de la aplicación que contiene todas las opciones de configuración del sitio. 3. Ir a Administrar→Configuración de sitio →Módulos. 4. deselecciona el módulo <i>“deliver-and-test”</i>. 5. Pulsar el botón “guardar”
------------------	---

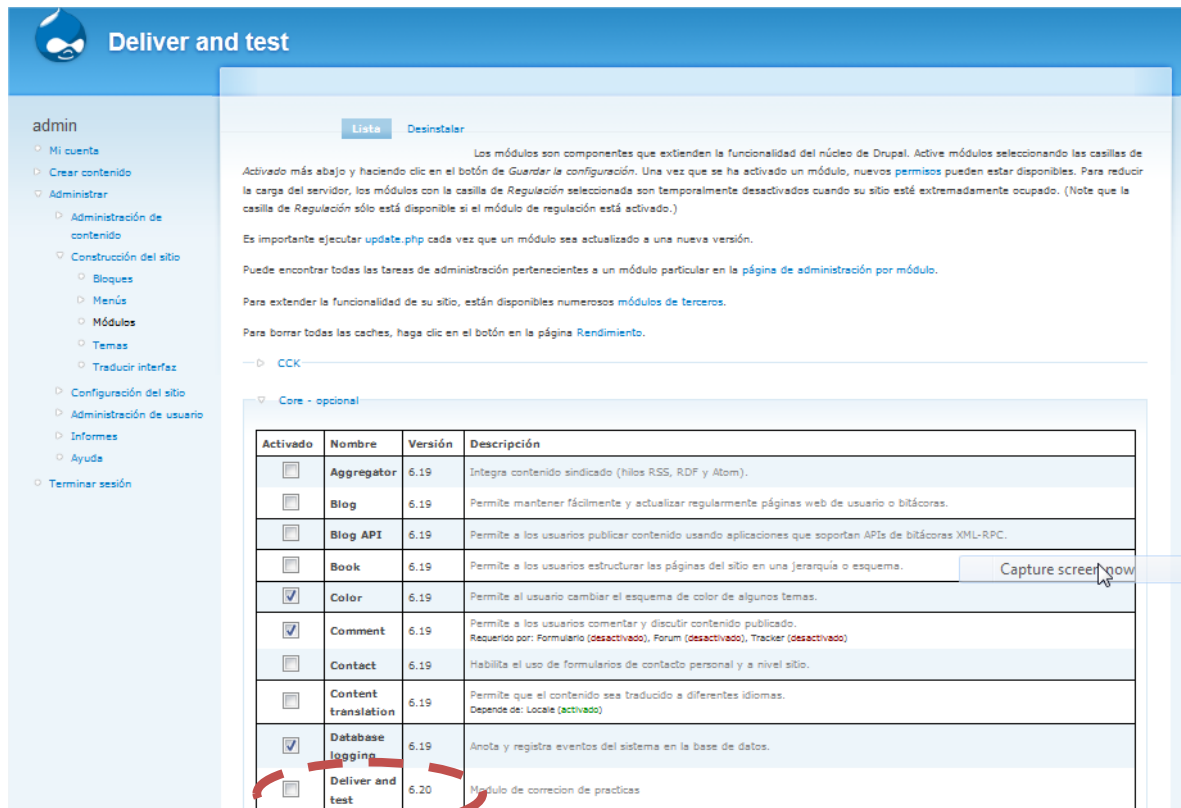


Figura 8: pantalla de desactivar module

3.1.4 Desinstalar módulo

Permitir borrar el módulo de la instancia de Drupal

Definición	El administrador desinstala el módulo <i>“deliver-and-test”</i> después de autenticarse correctamente
Usuarios	Con rol de administrador
Resultado	Después de autenticarse correctamente el administrador visualiza la página de bienvenida de drupal y el menú de configuración. Accediendo a la parte de módulo desactiva el módulo <i>“deliver-and-test”</i> y luego lo desinstala
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol administrador 2. Visualizar el menú en el lado Izquierdo de la aplicación que contiene todas las opciones de configuración del sitio. 3. Ir a Administrar→Configuración de sitio →Módulos. 4. Deseleccionar el módulo <i>“deliver-and-test”</i>. 5. Pulsar el botón “guardar”. 6. Ir a la pestaña “Desinstalar” que está en la parte superior de la página. 7. Seleccionar el módulo. 8. Pulsar el botón “guardar”.

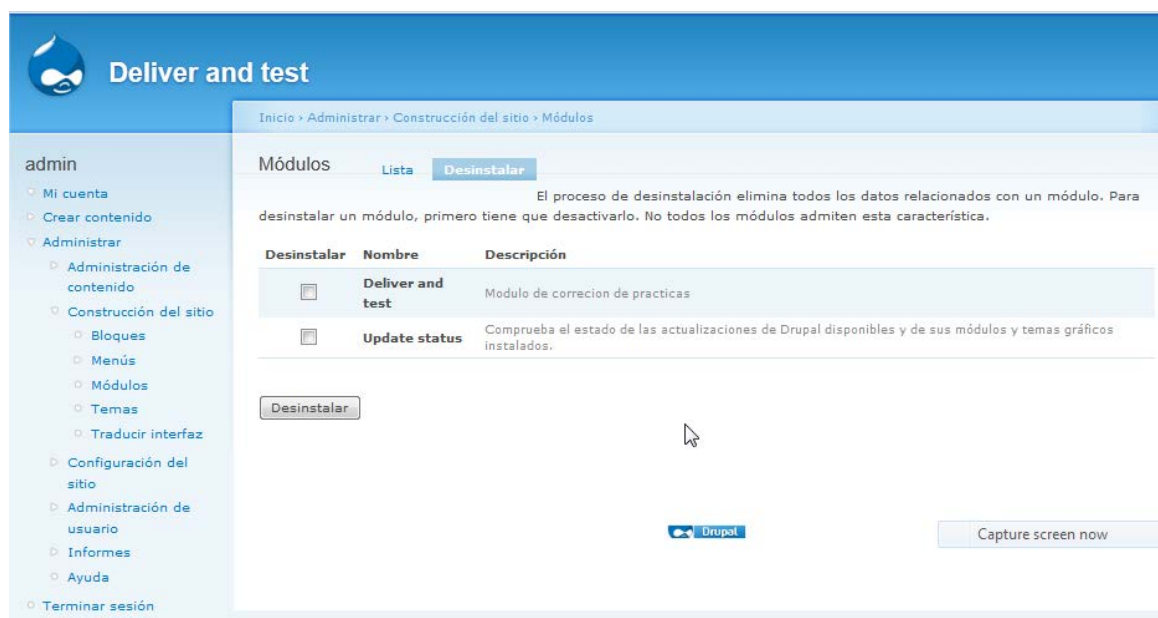



Figura 9: pantalla de desinstalar module

3.1.5 Modificar rol y datos del usuario

Se modifica el rol de usuario autenticado y sus datos personales.

Definición	El administrador modifica los datos de los usuario registrados
Usuarios	Con rol de administrador
Resultado	Después de autenticarse correctamente el administrador visualiza la página de bienvenida de Drupal y el menú de configuración. Accediendo a la parte de configuración de usuario para modificar sus datos y su rol.
Escenario	<ol style="list-style-type: none">1. Autenticarse con usuario con rol administrador2. Visualizar el menú en el lado Izquierdo de la aplicación que contiene todas las opciones de configuración del sitio.3. Ir a Administrar→Administración de usuario →usuarios.4. Seleccionar un usuario5. Modificar los datos del usuario.<ol style="list-style-type: none">5.1. Modificar los datos del usuario: se modifican el nombre, usuario y contraseña del usuario5.2. Modificar el rol: se selecciona uno de los roles que aparecen en el formulario de los datos del usuario.6. Pulsar el botón “guardar”



Deliver and test

admin

- Mi cuenta
- Crear contenido
- Administrar
 - Administración de contenido
 - Construcción del sitio
 - Configuración del sitio
 - Administración de usuario
 - Opciones de usuario
 - Permisos
 - Reglas de acceso
 - Roles
 - Usuarios
 - Informes
 - Ayuda
 - Terminar sesión

Lista

Agregar usuario

Drupal permite a los usuarios registrarse, conectarse, desconectarse, mantener sus perfiles de usuario, etc. Los usuarios del sitio no pueden usar sus propios nombres para publicar contenido mientras no hayan solicitado una cuenta de usuario.

[más ayuda...]

Mostrar sólo los usuarios en los que

☐ rol
 ☐ permiso
 ☐ estado

es

alumno

administrar bloques

activo

Filtrar

Actualizar opciones

Desbloquear a los usuarios seleccionados

Actualizar












 Usuario	Estado	Roles	Miembro durante	Último acceso	 Capture screen now
 wafaa	activo	 alumno	4 semanas 3 días	hace 1 semana 6 días	editar
 leonor	activo	 profesor	11 semanas 4 días	hace 11 semanas 4 días	editar
 luis	activo	 profesor	11 semanas 4 días	hace 1 semana 6 días	editar
 beatriz	activo	 alumno	11 semanas 4 días	hace 29 mins 38 segs	editar
 admin	activo		11 semanas 4 días	hace 1 min 40 segs	editar

Figura 10: Pantalla de usuarios registrados

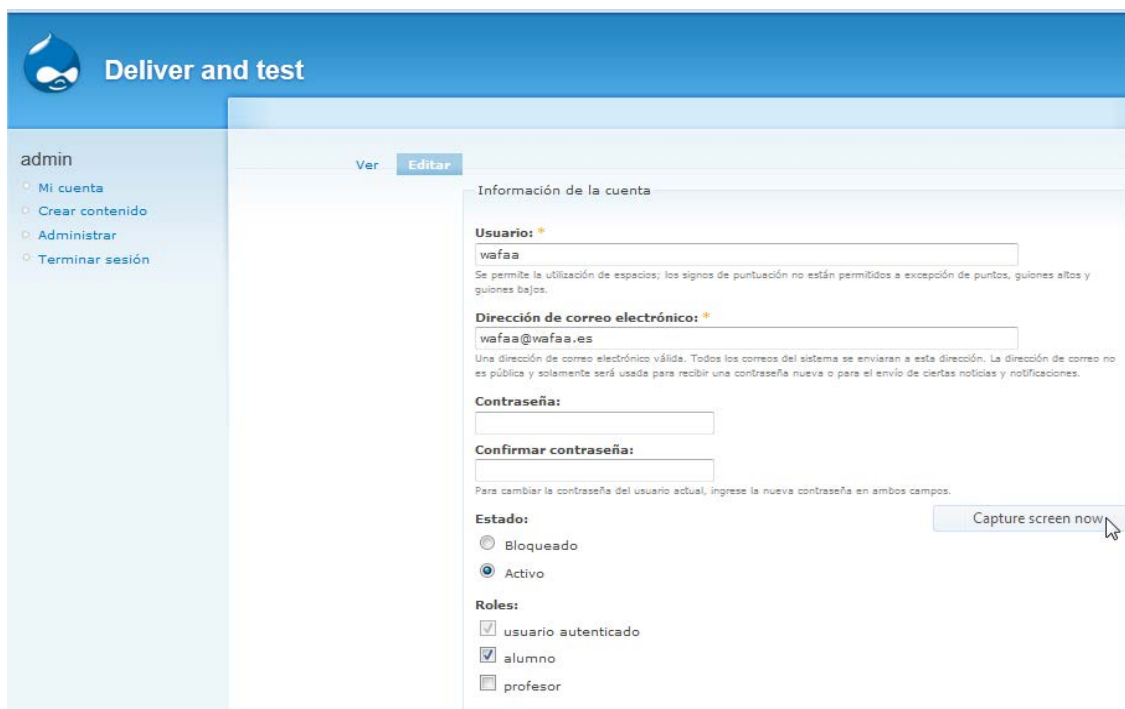


Figura 11: Pantalla de información de la cuenta del usuario registrado

3.1.6 Eliminar usuario

Se elimina el usuario de Drupal.

Definición	El administrador elimina un usuario registrados
Usuarios	Con rol de administrador
Resultado	Después de autenticarse correctamente el administrador visualiza la página de bienvenida de drupal y el menú de configuración. Accede a la parte de configuración de usuario para eliminar el usuario elegido.
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol administrador 2. Visualizar el menú en el lado Izquierdo de la aplicación que contiene todas las opciones de configuración del sitio. 3. Ir a Administrar→Administración de usuario →usuarios. 4. Seleccionar un usuario 5. Pulsar el botón "eliminar"

Figura 12: Pantalla de eliminar usuario

3.2 Alumno

3.2.1 Página principal

La página principal es la primera página que visualiza el alumno nada más autenticarse.

Definición	El alumno accede a su página principal después de autenticarse correctamente
Usuarios	Con rol de alumno
Resultado	Después de autenticarse correctamente el alumno visualiza todos los test , sus enunciados, sus interfaces y la última versión de la práctica que pasó el test con su nota, comentario y su

	aprobado si ha subido alguna.
Escenario	<p>3. Autenticarse con usuario con rol alumno</p> <p>4. Visualizar</p> <p>4.1. listado de test, su enunciado y su interfaz creados en por los profesores</p> <ul style="list-style-type: none"> Si el alumno tiene practicas subidas <p>Se muestran la última versión de la practica con los siguiente campos</p> <ul style="list-style-type: none"> ➤ Test: test elegido para validar la practica ➤ Enunciado: enunciado del test. ➤ Interfaz: interfaz del test. ➤ Practica: la última versión de la práctica subida. ➤ Aprobado: indica si pasó el test o no. ➤ Nota: La nota del profesor después de revisar el código de la práctica, si todavía no está corregida el valor por defecto de la nota es "--". ➤ Comentario: si el profesor ha puesto algún comentario sobre la práctica. ➤ Resultado: si hay errores de compilación o de ejecución se muestran en este campo ➤ Mensaje informativo si ha habido un cambio en el test. ➤ Botón "Añadir versión" ➤ Botón "Histórico" <ul style="list-style-type: none"> Si no tiene practicas subidas <ul style="list-style-type: none"> ➤ Solo aparece la pantalla con los test, sus interfaces y sus enunciados subidos por los profesores, el resto de los campos están vacíos. ➤ Un botón "Añadir práctica"

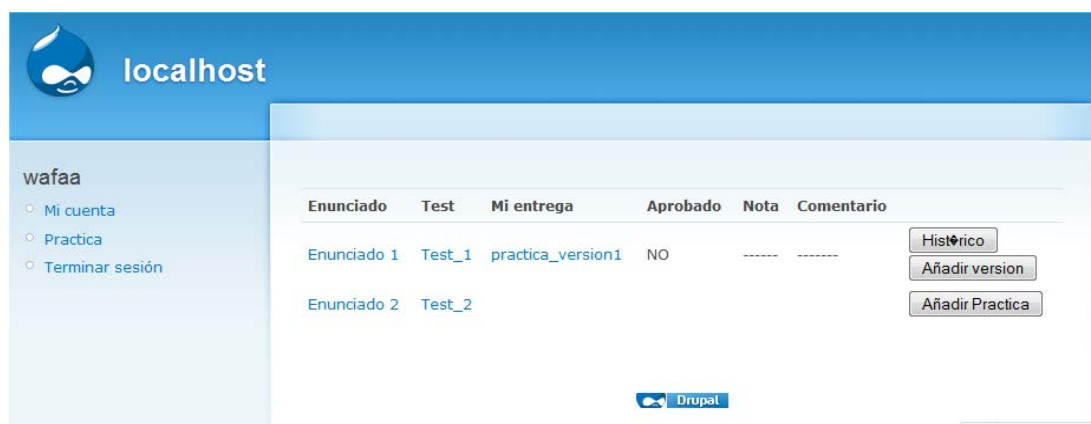


Figura 13: Pantalla de página principal del alumno sin **test** modificado

Deliver and test

beatriz

- Mi cuenta
- Practica
- Terminar sesión

Enunciado	Test	Interfaz	Mi entrega	Aprobado	Nota	
fase de version	TestHolaMundo	Saludo	HolaMundo_2	Si	6	<input type="button" value="Histórico"/> <input type="button" value="Añadir version"/>
Resultado	EJECUCION EXITOSA :array (0 => 'Se esperaba, "Hola Mundo"', 1 => 'Se obtuvo: Hola mundo', 2 => 'Valor de salida correcto. Test superado', 3 => 'ERROR',)					
Comentario	Proqueeeeeee					

Se ha modificado el test, El campo comentario muestra el resultado del cambio

Figura 14: Pantalla de página principal del alumno con **test** modificado

3.2.2 Añadir Practica

Un formulario para añadir la práctica del alumno.

Definición	El alumno accede a formulario para subir su practica
Usuarios	Con rol de alumno
Resultado	Al pulsar el botón “Añadir práctica” aparece un formulario con campos que hay que rellenar para añadir una practica
Escenario	<ol style="list-style-type: none"> Autenticarse con usuario con rol alumno Visualizar página principal Pulsar el botón “Añadir práctica” Visualizar un formulario con un campo que hay que rellenar <ol style="list-style-type: none"> Los campos del formulario son <ul style="list-style-type: none"> Practica: path de la practica Test: no editable solo muestra el dato del test Enunciado: link del enunciado del test. Al final del formulario se muestran 2 botones “Subir práctica” y “Volver”

Figura 15: Pantalla del formulario añadir Practica

3.2.3 Añadir versión

Un formulario para añadir una nueva versión de la práctica.

Definición	El alumno accede a un formulario para subir versión nueva de la práctica.
Usuarios	Con rol de alumno
Resultado	Al pulsar el botón "Añadir versión" aparece un formulario con campos que hay que rellenar para añadir la versión una practica
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Pulsar el botón "Añadir versión" 4. Visualizar un formulario con una serie de campos que hay que rellenar <ol style="list-style-type: none"> 4.1. Los campos del formulario son <ul style="list-style-type: none"> • Nombre de la práctica: el nombre de la práctica originar con la versión, campo no editable • Test: el test que pasa la práctica que es el mismo para todas las versiones(CAMPO NO EDITABLE) • Enlace: después del test aparece un enlace para descargar el enunciado del test. • Practica a subir: :subir el archivo <p>Al final del formulario se muestran 2 botones:</p> <p>"Subir practica": Guarda los datos en base de datos</p> <p>"Volver": Vuelve a la página anterior sin guardar ningún dato</p>


Figura 16 : Pantalla del formulario “Añadir versión”

3.2.4 Histórico

Pantalla que muestra todas las versiones de la práctica seleccionada, subidas por el alumno

Definición	El alumno visualiza todas las versiones de la práctica seleccionada
Usuarios	Con rol de alumno
Resultado	Al pulsar el botón “Histórico” aparece todas las versiones de la practica
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Pulsar el botón “Histórico” 4. Se visualiza todas las versiones de la practica con los siguientes campos <ul style="list-style-type: none"> ✓ Versiones de práctica: el nombre de la versión ✓ Aprobado: indica si la práctica está aprobada ✓ Nota: muestra la nota, si el profesor ha puntuado, de la práctica. ✓ Comentario: muestra el comentario agregado por el profesor. ✓ Resultado: muestra el mensaje de error si se ha producido error en compilación o ejecución. <p>Al final de la página se muestra 1 botones:</p>

"Volver": Vuelve a la página anterior sin guardar ningún dato



Deliver and test

beatriz

- Mi cuenta
- Practica
- Terminar sesión

versiones de la practica	Aprobado	Resultado	Nota	Comentario
HolaMundo_1	NO	ERROR DE EJECUCIÓN array (0 => 'Se esperaba, "Hola Mundo"', 1 => 'Se obtuvo: Hola Mundo', 2 => 'Valor de salida incorrecto :(. Test no superado', 3 => 'ERROR',)	0	
HolaMundo_2	Si	EJECUCION EXITOSA :array (0 => 'Se esperaba, "Hola Mundo"', 1 => 'Se obtuvo: Hola mundo', 2 => 'Valor de salida correcto. Test superado', 3 => 'ERROR',)	6	Proqueeeeeee

Volver

Capture screen now

Figura 17 : Pantalla de "Histórico"

3.2.5 Descargar test

El alumno tiene la opción de descargar el **test** y ver su contenido

Definición	El alumno descarga el test
Usuarios	Con rol de alumno
Resultado	Al pulsar el nombre del test se abre la pop up que permite descargarlo
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Pulsar el nombre del test 4. Aparece la ventana de descarga que permite guardar o abrir el test

3.2.6 Descargar enunciado

El alumno tiene la opción de descargar el enunciado y ver su contenido

Definición	El alumno descarga el enunciado
Usuarios	Con rol de alumno
Resultado	Al pulsar el nombre de enunciado se abre una ventana que permite descargarlo
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Pulsar el nombre del enunciado 4. Aparece la ventana de descarga que permite guardar o abrir el enunciado.

3.2.7 Descargar interfaz

El alumno tiene la opción de descargar la interfaz y ver su contenido

Definición	El alumno descarga la interfaz
Usuarios	Con rol de alumno
Resultado	Al pulsar el nombre de enunciado se abre una ventana que permite descargarlo
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Pulsar el nombre de la interfaz 4. Aparece la ventana de descarga que permite guardar o abrir el enunciado.

3.2.8 Descargar mi entrega

El alumno tiene la opción de descargar su práctica y ver su contenido

Definición	El alumno descarga su entrega
Usuarios	Con rol de alumno
Resultado	Al pulsar el nombre de la práctica se abre una ventana que permite descargarla
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Pulsar el nombre de la practica 4. Aparece la ventana de descarga que permite guardar o abrir la practica

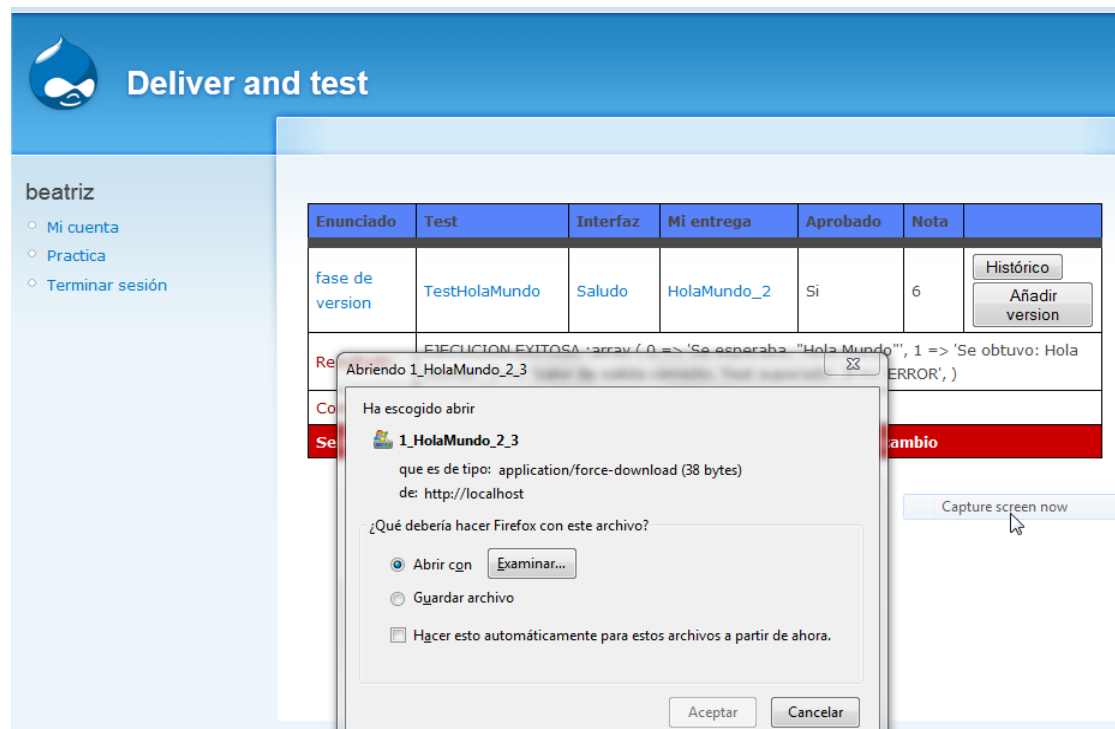


Figura 18: pantalla de descargar un fichero

3.2.9 Notificar la modificación del test

El alumno recibe un mensaje notificando la modificación del **test** realizada por el profesor que lo ha colgado.

Definición	Se avisa al alumno del cambio de test realizado por el profesor
Usuarios	Con rol de alumno
Resultado	Se muestra un mensaje, debajo del test en la página principal, indicando al alumno el cambio realizado por el profesor
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol alumno 2. Visualizar página principal 3. Se muestran nuevas evaluaciones <ol style="list-style-type: none"> 3.1. El campo aprobado modificado 3.2. La nota se eliminar hasta que vuelva a evaluar el profesor 3.3. Se agrega en el campo resultado el resultado de la compilación/Ejecución si existe algún error. 4. Visualizar debajo del test modificado un mensaje notificando el cambio.

3.3 Profesor

3.3.1 Página principal

La página principal es la primera página que visualiza el profesor nada más autenticarse.

Definición	El profesor accede a su página principal después de autenticarse correctamente
Usuarios	Con rol de profesor
Resultado	Después de autenticarse correctamente el profesor visualiza un listado de test con sus enunciados, fecha de inicio y fecha fin, si ha subido alguna, sino se muestra la tabla vacía
Escenario	<p>1. Autenticarse con usuario con rol profesor</p> <p>1.1. Se visualiza listado de test con los siguientes campos</p> <ul style="list-style-type: none"> ✓ Test ✓ Enunciado ✓ Interfaz ✓ Fechas de inicio: fecha de alta del test ✓ Fecha fin: fecha de finalización del test <p>Con los botones ver todos test, añadir test, practicas, Modificar y eliminar</p>

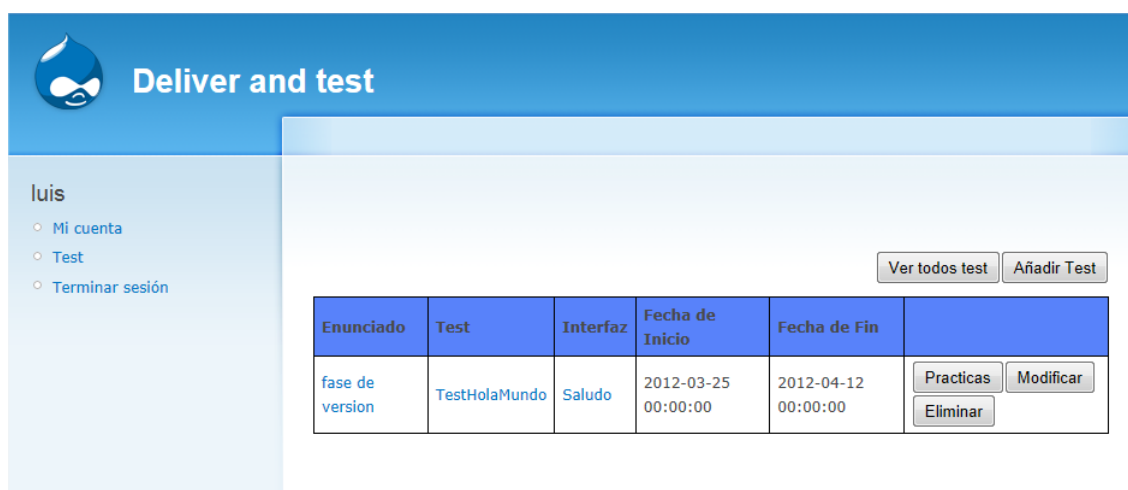
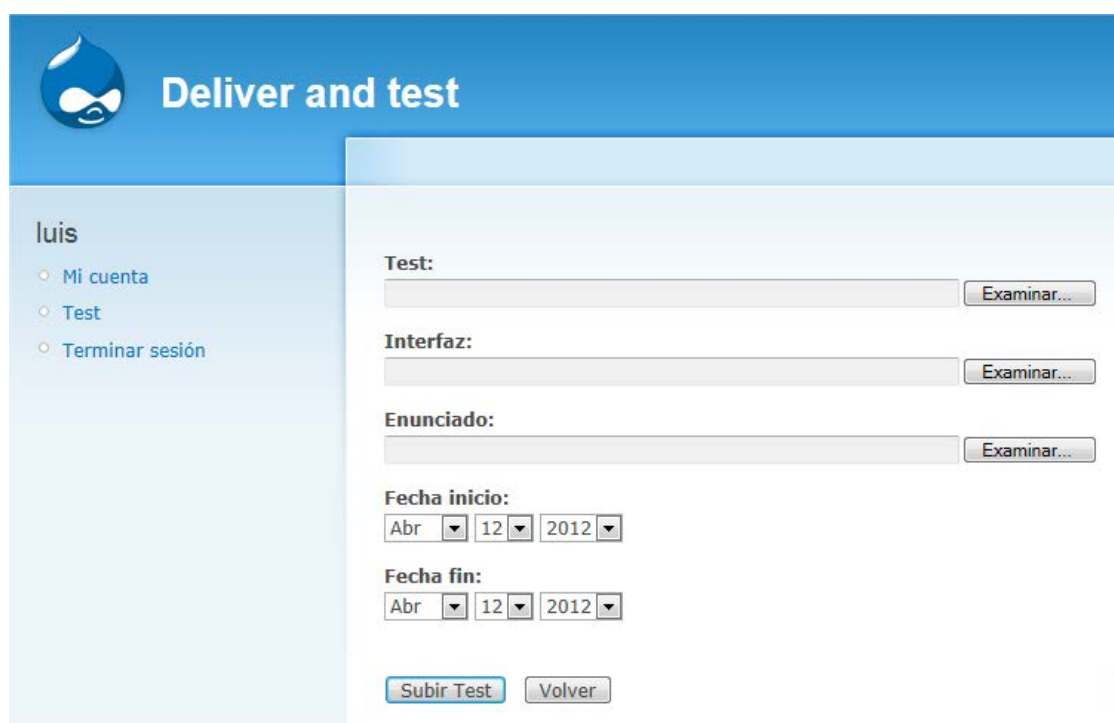


Figura 19: Pantalla de “Página principal”

3.3.2 Añadir test

Un formulario para añadir un **test**.

Definición	El profesor añade un test nuevo
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón "Añadir versión" aparece un formulario con campos que hay que rellenar para añadir la versión una practica
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón "Añadir test" 4. Visualizar un formulario con una serie de campos que hay que rellenar <ol style="list-style-type: none"> 4.1. Los campos del formulario son <ul style="list-style-type: none"> • Nombre: el nombre del test • Fecha inicio: fecha de subida • Fecha fin: fecha finalización del test • Test: el test que usan los alumnos para validar sus practicas • Nombre enunciado: Nombre del enunciado • Enunciado: Enunciado del test explicativo. <p>Al final del formulario se muestran 2 botones:</p> <p>"Subir test": Guarda los datos en base de datos.</p> <p>"Volver": Vuelve a la página anterior sin guardar ningún dato</p>



Deliver and test

luis

- Mi cuenta
- Test
- Terminar sesión

Test:

Interfaz:

Enunciado:

Fecha inicio:

Abr ▼ 12 ▼ 2012 ▼

Fecha fin:

Abr ▼ 12 ▼ 2012 ▼

Figura 20 : Pantalla del formulario "Añadir test"

3.3.3 Descargar test

El profesor tiene la opción de descargar un test y ver su contenido

Definición	El profesor descarga el test
Usuarios	Con rol de profesor
Resultado	Al pulsar el nombre del test se abre una ventana que permite descargarlo
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el nombre del test 4. Aparece la ventana de descarga que permite guardar o abrir el test

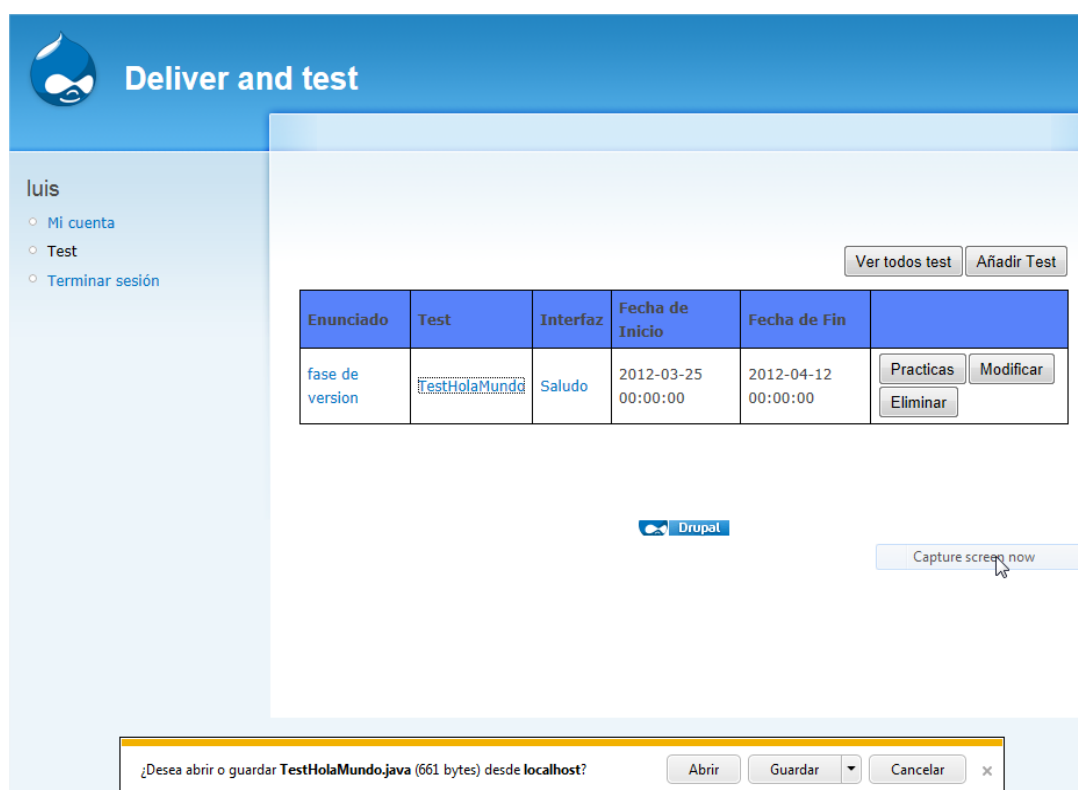


Figura 21 : Pantalla de "Descargar test"

3.3.4 Descargar enunciado

El profesor tiene la opción de descargar un enunciado de un test y ver su contenido

Definición	El profesor descarga el enunciado
Usuarios	Con rol de profesor
Resultado	Al pulsar el nombre de enunciado se abre una ventana que permite descargarlo
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el nombre del enunciado 4. Aparece la ventana de descarga que permite guardar o abrir el enunciado

3.3.5 Descargar interfaz

El profesor tiene la opción de descargar un interfaz de un test y ver su contenido

Definición	El profesor descarga la interfaz
Usuarios	Con rol de profesor
Resultado	Al pulsar el nombre de la interfaz se abre una ventana que permite descargarlo
Escenario	<ol style="list-style-type: none"> 5. Autenticarse con usuario con rol profesor 6. Visualizar página principal 7. Pulsar el nombre de la interfaz 8. Aparece la ventana de descarga que permite guardar o abrir la interfaz

3.3.6 Practicas

El profesor visualiza todas las prácticas que pasaron un test elegido de los test que ha cargado.

Definición	El profesor visualiza todas las practicas que han pasado el test
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón "Practicas" aparece un listado de practicas
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón "Practicas" 4. Aparece un listado de prácticas, que pasaron el test elegido, con los siguientes campos: <ol style="list-style-type: none"> 4.1. Práctica. Link con el nombre de la practica 4.2. Aprobado: indica si la práctica esta validada 4.3. Nota: la nota que pone el profesor. 5. se muestran 2 botones: <ul style="list-style-type: none"> "Histórico": Muestra todas las versiones de la práctica ver el caso de uso [Ver histórico]. "modificar Nota": Modificar la nota de la práctica. Ver el caso de uso [Modificar nota]. 6. Al final de la página aparece un botón "Volver" que permite volver a la página anterior.

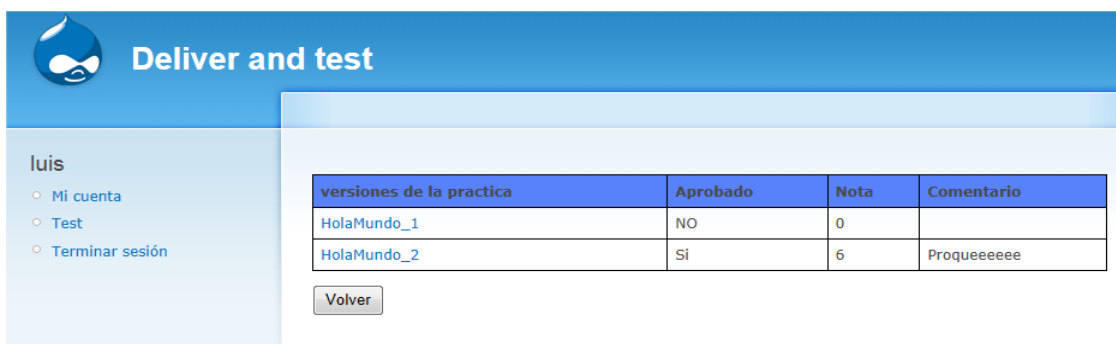


Figura 22 : Pantalla de “Practicas”

3.3.7 Ver histórico

Pantalla que muestra todas las versiones de la práctica seleccionada, subidas por el alumno

Definición	El profesor visualiza todas las versiones de la práctica elegida
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón “Histórico” aparece un listado de todas las versiones de las practicas
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón “Practicas” 4. Aparece un listado de prácticas, que pasaron el test elegido. 5. Pulsar el botón “Histórico” de una practica 6. Se muestra un listado de todas las versiones de la práctica seleccionada, con los siguientes campos: <ol style="list-style-type: none"> 6.1. Versión de la práctica. Link con el nombre de la practica 6.2. Aprobado: indica si la práctica esta validada 6.3. Nota: la nota que pone el profesor 7. Al final de la página aparece un botón “Volver” que permite volver a la página anterior.



The screenshot shows a web interface titled "Deliver and test". On the left, there is a sidebar with the user's name "luis" and three menu items: "Mi cuenta", "Test", and "Terminar sesión". The main content area displays a table with the following data:

versiones de la practica	Aprobado	Nota	Comentario
HolaMundo_1	NO	0	
HolaMundo_2	Si	6	Proqueeeeeee

Below the table, there is a button labeled "Volver".

Figura 23: Pantalla de "Histórico" de las practicas

3.3.8 Modificar nota

Formulario que permite al profesor puntual una práctica de un alumno.

Definición	El profesor asigna una nota al alumno después de revisar su practica
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón "Modificar nota" se muestra un formulario que permite asignar nota al alumno
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón "Practicas" 4. Aparece un listado de prácticas, que pasaron el test elegido. 5. Pulsar el botón "Modificar nota "de una practica 6. Se muestra un formulario con los siguientes campos: <ol style="list-style-type: none"> 6.1. Nota: la nota que quiere asignar el profesor 6.2. Comentario: Permite al profesor dejar algún comentario para el alumno 7. Al final de la página aparece 2 botones: <ol style="list-style-type: none"> 7.1. "Modificar nota": Guarda los datos en la base de datos. 7.2. "Volver": permite volver a la página anterior.

Figura 24: Pantalla del formulario “Modificar nota”

3.3.9 Modificar test

Formulario que permite al profesor modificar su test

Definición	El profesor modifica un test
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón “Modificar test” se muestra un formulario que hay que rellenar para modificar el test subido
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón “Modificar test” de un test del listado. 4. Se muestra un formulario con los siguientes campos: <ol style="list-style-type: none"> 4.1. test: fichero del test Nuevo 4.2. enunciado: el link con el nombre del enunciado 4.3. interfaz. 5. Al final de la página aparecen 2 botones: <ol style="list-style-type: none"> 5.1. “Modificar test”: Guarda los datos en la base de datos “Volver”: permite volver a la página anterior. 6. Si se pulsa el botón “Modificar test”, aparte de modificar el test se compila y se ejecuta de nuevo todas las prácticas que han elegido este test. Eso provoca una nueva evaluación de las prácticas notificando el cambio a los alumnos a los cuales pertenecen las prácticas.

Figura 25: Pantalla del formulario “Modificar test”

3.3.10 Eliminar test

El profesor podrá eliminar el test de forma ‘lógica’, sin borrarlo en base de datos.

Definición	El profesor elimina un test
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón “eliminar” se elimina el test de forma lógica
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón “eliminar” de un test del listado. 4. Eliminar el test de la listado de test

3.3.11 Ver todos los test

El profesor visualiza todos los test tanto los borrados como los activos

Definición	El profesor visualiza los test
Usuarios	Con rol de profesor
Resultado	Al pulsar el botón “eliminar” se elimina el test de forma lógica
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón “Ver todos los test”. 4. Se muestran todos los test que ha subido el profesor, los que están borrados y los que no.



Figura 26: Pantalla de “ver todos los test”

3.3.12 Modificar fecha fin

El botón “Modificar fecha fin” permite modificar la fecha máxima que tiene el test.

Definición	El profesor modifica la fecha fin de su test
Usuarios	Con rol de profesor
Resultado	Al pulsar “Modificar fecha fin” aparece un formulario que permite la modificación de la fecha fin
Escenario	<ol style="list-style-type: none"> 1. Autenticarse con usuario con rol profesor 2. Visualizar página principal 3. Pulsar el botón “Ver todos los test”. 4. Se muestran todos los test que ha subido el profesor, los que están borrado y los que no. 5. Pulsar el botón “Modificar fecha fin”. 6. Se muestra un formulario que permite modificar la fecha fin del test.



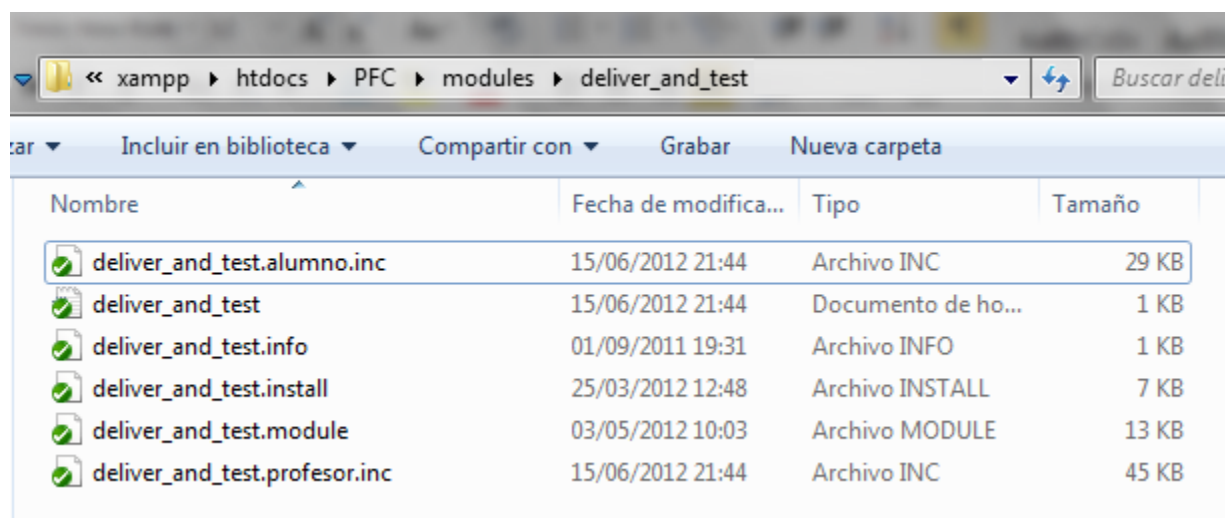
Figura 27: Pantalla del formulario “Modificar fecha fin”

4 Implementación

4.1 Creación del modulo

Después de instalar Drupal se crea el módulo “**Deliver_and_Test**” en la carpeta de módulos de Drupal “modules”[Figura 28: Directorio del módulo “Deliver-and-test”

]



Nombre	Fecha de modifica...	Tipo	Tamaño
deliver_and_test.alumno.inc	15/06/2012 21:44	Archivo INC	29 KB
deliver_and_test	15/06/2012 21:44	Documento de ho...	1 KB
deliver_and_test.info	01/09/2011 19:31	Archivo INFO	1 KB
deliver_and_test.install	25/03/2012 12:48	Archivo INSTALL	7 KB
deliver_and_test.module	03/05/2012 10:03	Archivo MODULE	13 KB
deliver_and_test.profesor.inc	15/06/2012 21:44	Archivo INC	45 KB

Figura 28: Directorio del módulo “Deliver-and-test”

Dentro de esta carpeta se crean e implementan los ficheros “*inc*” que son ficheros donde se abarca todos los métodos que contienen las funcionalidades. Para cada rol que existe, profesor o alumno, se crea un fichero “*.inc*” con las implementaciones representativas que le corresponde. El fichero “*.install*” instala y desinstala las tablas que pertenecen al modelo de datos del proyecto e inserta en la tabla Rol propia de *Drupal* los 2 nuevos rol de la aplicación, profesor/alumno. El fichero “*.module*” contiene la implementación que define la ruta y los permisos de los métodos definidos en los ficheros “*.inc*” y por último el fichero “*.info*” es un documento con metadatos que contiene la versión de *Drupal*, el autor, etc.

A continuación se explican los detalles del contenido de los ficheros indicados anteriormente:

- **deliver_and_test.alumno.inc:** Fichero que contiene todo el desarrollo de las funcionalidades del alumno. El fichero contiene los siguientes métodos:
 - **deliver_and_test_index:** Cuando el alumno se autentica, se llama a esta función que muestra la página principal con los datos principales al que puede tener acceso, enunciado, test, interfaz, la practica (mi entrega),

aprobado, nota y los botones “Añadir práctica”, “Histórico” e “Añadir version” Hay campos que se muestran solo si se producen las siguientes acciones:

- si existe por lo menos una práctica subida se muestran comentario, resultado y los botones “añadir version” y “Histórico” con los valores de salida de ejecución de la función ***deliver_and_test_compilarTestP***.
 - Si el profesor modifica un test se notifica a los alumnos, que han elegido el test modificado para evaluar su práctica, con un mensaje informativo que se muestra justo por debajo del campo comentario.
-
- **deliver_and_test_anadirPractica:** formulario que permite al alumno añadir una nueva y primera version de la práctica. Después de rellenar el formulario se guardan los datos insertados en su tabla correspondiente y vuelve a la página principal.
 - **deliver_and_test_salida:** Muestra la salida de la compilación y ejecución cuando se ejecuta la función ***deliver_and_test_compilarTestP***.
 - **deliver_and_test_anadirVersion:** formulario que permite añadir una nueva versión de una práctica ya agregada. Los datos insertados se guardan en su tabla correspondiente y vuelve a la página principal.
 - **deliver_and_test_compilarTestP:** Es la función que compila y ejecuta la práctica, cuando se añade una práctica, y devuelve el resultado y el mensaje de salida de la ejecución. Estos valores devueltos son las *Nota* y *Resultado*
 - **deliver_and_test_verPracticasAlumno:** muestra todas las prácticas del alumno autenticado junto con los campos aprobado, resultado, nota y comentario.
 - **deliver_and_test_obtenerVersionPractica:** devuelve la versión de la última práctica insertada para asignar automáticamente el nombre de la siguiente versión del alumno para que no haya confusión ni errores a la hora de subir la version de la práctica.
 - **deliver_and_test_descargarFicheroAlumno:** Descarga el fichero seleccionado.
 - **deliver_and_test_nombre:** devuelve el nombre de la práctica sin extensión.
 - **deliver_and_test_ultimaPosicionPractica:** Devuelve la última version de la practica

- **deliver_and_test.profesor.inc:** Fichero que contiene todo el desarrollo de las funcionalidades del profesor. El fichero contiene los siguientes métodos:
 - **deliver_and_test_index:** Cuando se autentica el profesor , se llama a esta función que muestra la página principal con los datos principales al que puede tener acceso, enunciado, test, interfaz, fecha inicio, fecha fin y los botones “ver todos test”, “añadir test”, “practica”, “modificar” y “eliminar”.
 - **deliver_and_test_eliminarTest:** cuando se pulsa el botón “Eliminar” se ejecuta esta función para eliminar de forma lógica el test elegido.
 - **deliver_and_test_verTodosTest:** Cuando se pulsa el botón “ver todos test” se ejecuta esta función que muestra todos los test existentes en la base de datos y que pertenecen al profesor autenticado.
 - **deliver_and_test_modificarTest:** cuando se pulsa el botón “Modificar” se muestra un formulario de modificación test que permite cambiar los campos test, enunciado e interfaz y guardarlos en su tabla correspondiente.
 - **deliver_and_test_modificarNota:** cuando visualiza el profesor todas las prácticas, que han superado o no el test elegido, se permite acceder al formulario de puntuación a través del botón “Modificar nota”. Este formulario contiene, aparte del campo nota, un campo comentario que permite al profesor poner cualquier comentario sobre la práctica revisada. Los datos se guardan en su tabla correspondiente.
 - **deliver_and_test_anadirTest:** cuando se pulsa el botón “Añadir test” se muestra un formulario para agregar un test nuevo junto a su enunciado e interfaz. Cuando se insertan los datos en el formulario se guardan en su tabla correspondiente.
 - **deliver_and_test_selectIdTest:** Obtiene el identificador de un test específico.
 - **deliver_and_test_insertTestPorfesor:** inserta la relación test-profesor en la tabla correspondiente en la base de datos.
 - **deliver_and_test_borrarTest:** Se ejecuta siempre cuando se autentica el profesor para eliminar todos los test que tienen la fecha fin anterior a la fecha del sistema.
 - **deliver_and_test_verPracticasProfesor:** cuando se pulsa el botón “practicas” se ejecuta esta función para mostrar todas las prácticas que han elegido el test seleccionado.
 - **deliver_and_test_verPracticasAlumnoP:** cuando se selecciona el botón “Histórico” de la pantalla que se muestra cuando se ejecuta la función anterior “*deliver_and_test_verPracticasProfesor*” Obtiene todas las versiones de la práctica del alumno seleccionado.

- **deliver_and_test_descargarFicheroProfesor:** Descarga el fichero seleccionado.
 - **deliver_and_test_modificarFechaFin:** cuando se muestran todos los test subidos por el profesor autenticado se muestra un botón “Modificar fecha fin” que es un formulario para modificar la fecha fin del test seleccionado. Permite al profesor ampliar o reducir la fecha de entrega.
 - **deliver_and_test_compilarTestP :** cuando un profesor modifica su test se ejecuta esta función para compilar, ejecutar todas las prácticas que han elegido el test modificado y devuelve los resultados de las operaciones anteriores y los guarda en la base de datos.
 - **deliver_and_test_nombre:** devuelve los nombres del test sin extensión
- **deliver_and_test.module:** Donde se definen las reglas de acceso. El fichero contiene los siguientes método:
 - **deliver_and_test_form_alter :** permite alterar los formularios propios de drupal como por ejemplo el formulario de registro. En este proyecto se utiliza esta función para añadir en el formulario de registrarse la opción de rol, profesor o alumno, para que un usuario pueda elegir su rol al registrarse.
 - **function deliver_and_test_user:** permite acceder a la página principal según rol, es decir, si el autenticado es un alumno aparece la página principal del alumno y si es profesor aparece la página principal del profesor.
 - **function deliver_and_test_links :** define los enlaces del menu.
 - **function deliver_and_test_menu :** se definen los nombre y las acciones correspondientes al nombre y la clase al que corresponde, por ejemplo, se define la página principal del profesor con el nombre *index*, que llama a la función **deliver_and_test_index** de la clase **deliver_and_test.profesor.inc**.
 - **function deliver_and_test_user_register_submit**
 - **function deliver_and_test_init:** añade un estilo personalizado.

A continuación se muestran una parte de las funciones definidas en la función menú de la clase module:

```
function deliver_and_test_menu() {  
  global $user;  
  $usuario=$user->roles;  
  $items['menuTest'] = array(  
    'title' => 'Test',  
    'page callback' => 'deliver_and_test_index',  
    'page arguments' => "",  
    'access arguments' => array('profesor'),  
    'access callback' => 'profesor_module_access',  
  );  
}
```

```

    'type' => MENU_NORMAL_ITEM,
    // 'access' => user_access('profesor',$usuario),
    'file' => 'deliver_and_test.profesor.inc',
  );
  return $items;
}

```

- **deliver_and_test.install**: se crean las tablas, propias del módulo, en la base de datos Drupal. Esta clase se ejecuta cuando el administrador activa el módulo.

El fichero contiene los siguientes métodos:

- o **function deliver_and_test_schema**: se definen todas las tablas propias especificando los campos y el tipo de datos, en la base de datos de drupal.

Mostramos a continuación una tabla creada en la función **function deliver_and_test_schema**

```

//TEST
$schema['deliver_and_test_profesorTest'] = array(
  'description' => 'Tabla para test',
  'fields' => array
  (
    'id' => array (
      'description' => t('Identificador de profesor'),
      'type' => 'serial',
      'unsigned' => TRUE,
      'not null' => TRUE,
    ),
    'idProfesor' => array (
      'description' => t('id profesor'),
      'type' => 'int',
      'not null' => TRUE,
    ),
    'idTest' => array (
      'description' => t('id test'),
      'type' => 'int',
      'not null' => TRUE,
    ),
  ),
  'primary key' => array('id'),
);

```

- o **function deliver_and_test_install**: se ejecuta la creación de las tablas definidas en la función schema.

```

function deliver_and_test_install() {
  // Create my tables.
  drupal_install_schema('deliver_and_test');
  // Crear nuevos roles del nuestro modulo
  $params = _hook_get_roles_by_name ();
  array_unshift ($params, 'INSERT INTO {role} (name) VALUES (\'%s\')'. str_repeat ('',
  (\'%s\')', count ($params) - 1));
  // add roles to system
  call_user_func_array ('db_query', $params);}

```

- o **function deliver_and_test_uninstall:** Se desinstalan todas las tablas creadas en la función schema. Esta función se ejecuta cuando el administrador desactive el módulo en drupal.

```
function deliver_and_test_uninstall() {  
  
  // Drop my tables.  
  drupal_uninstall_schema('deliver_and_test');  
  $params = _hook_get_roles_by_name ();  
  //Borrar los roles nada más al desinstalar el modulo  
  array_unshift ($params, 'DELETE FROM {role} WHERE name = \'%s\' . str_repeat ('  
or name = \'%s\'", count ($params) - 1));  
  // delete roles to system  
  call_user_func_array ('db_query', $params);  
  array_unshift ($params, 'DELETE FROM {users_roles} WHERE name = \'%s\' .  
str_repeat ('\'%s\'", count ($params) - 1));  
  // delete roles to system  
  call_user_func_array ('db_query', $params);  
}
```

- o **function _hook_get_roles_by_name:** Permite agregar los roles, profesor y alumno, en el campo de roles propio de grupal.

```
//ROLES  
function _hook_get_roles_by_name() {  
  
  // Return role information  
  return array (  
    'profesor',  
    'alumno',  
  );  
}
```

- **deliver_and_test.info** : se definen los metadatos

```
; $Id: delived.info,v 1.4 2007/06/08 05:50:56 dries Exp $  
name = Deliver and test  
description = "Modulo de corrección de prácticas"  
package = Core - optional  
version = VERSION  
core = 6.x  
; Information added by drupal.org packaging script on 2010-08-11  
version = "6.20"  
project = "drupal"  
datestamp = "1281559292"  
regions[content_top] = content top  
regions[content_bottom] = content bottom
```

- **deliver_and_test.css:** se define el estilo personalizado de módulo creado.

4.2 Creación de las tablas en la base de datos

En este proyecto se ha optado por crear las tablas propias ya que de las tablas de Drupal solo reutilizaremos la tabla de **user** y **rol** para recuperar los datos del usuario autenticado y su rol.

En el fichero "**deliver_and_test.install**" se han creado las siguientes tablas:

- La tabla "**test**": es la tabla que contiene la información del test que sube el profesor autenticado. La tabla contiene los siguientes campos:
 - ✓ **IdTest**: identificador de test
 - ✓ **Nombre**: Nombre del test
 - ✓ **fecha_inicio**: fecha de subida del test
 - ✓ **fecha_fin**: fecha de fin del test
 - ✓ **pathTest**: path del test
 - ✓ **borrado**: borrado lógico del test cuando se pase la *fecha_fin*
 - ✓ **enunciado**: nombre del enunciado que explica con detalle el test
 - ✓ **pathEnunciado**: path del enunciado.
 - ✓ **Interfaz**: nombre de la interfaz que corresponde al test
 - ✓ **pathInterfaz**: path de la interfaz
 - ✓ **borrado**: indica si el test esta borrado o ha pasado su fecha final
 - ✓ **numCambioTest**: indica el número de cambios

```
//Tabla de Test
$schema['deliver_and_test_test'] = array(
  'description' => 'Tabla para test',
  'fields' => array
  (
    'idTest' => array
    (
      'description' => t('Identificador de profesor'),
      'type' => 'serial',
      'unsigned' => TRUE,
      'not null' => TRUE,
    ),
    'nombre' => array
    (
      'description' => t('Nombre del test'),
      'type' => 'varchar',
      'length' => '255',
      'not null' => TRUE,
    ),
    'fecha_inicio' => array
    (
      'description' => t('Fecha inicio de la entrega'),
```

```
'type' => 'datetime',
'not null' => FALSE,
),
),
'primary key' => array('idTest'),
);
```

- La tabla "**profesorTest**": contiene la relación de test-profesor. La información que contiene la tabla es:

- ✓ **Id**: identificador de la relación
- ✓ **idTest**: identificador del test subido
- ✓ **idProfesor**: identificador del profesor que acaba de subir el test

```
$schema['deliver_and_test_profesorTest'] = array(
'description' => 'Tabla para test',
'fields' => array
(
'id' => array
(
'description' => t('Identificador de profesor'),
'type' => 'serial',
'unsigned' => TRUE,
'not null' => TRUE,
),
'idProfesor' => array
(
'description' => t('id profesor'),
'type' => 'int',
'not null' => TRUE,
),
'idTest' => array
(
'description' => t('id test'),
'type' => 'int',
'not null' => TRUE,
),
),
'primary key' => array('id',));
```

- La tabla "**Práctica**": almacena toda la información de la práctica que sube el alumno desde la aplicación. La información que contiene la tabla es:

- ✓ **IdPractica**: identificador de la practica
- ✓ **idTest**: identificador del test
- ✓ **idAlumno**: identificador del alumno
- ✓ **nombrePractica**: nombre de la practica
- ✓ **notaDefinitiva**: nota que define el profesor
- ✓ **aprobado**: cuando se aprueba la práctica se inserta un "1" en este campo

- ✓ **pathPractica**: path de la practica

- ✓ **comentario:** Comentario del profesor sobre la práctica.
- ✓ **salidaEjecucion:** resultado de la ejecución.

```
//Practica
$schema['deliver_and_test_practica'] = array(
  'description' => 'Tabla para prácticas',
  'fields' => array
  (
    'id' => array
    (
      'description' => t('Identificador'),
      'type' => 'serial',
      'unsigned' => TRUE,
      'not null' => TRUE,
    ),
    'idPractica' => array
    (
      'description' => t('Identificador de practica'),
      'type' => 'int',
      'unsigned' => TRUE,
      'not null' => TRUE,
    ),
    'idAlumno' => array
    (
      'description' => t('id profesor'),
      'type' => 'int',
      'not null' => TRUE,
    ),
  ),
  'primary key' => array('id', );
```

Después de crear las tablas se agrega funciones de instalación y de desinstalación que se ejecutan cuando se activa/desinstala el módulo para crear/borrar las tablas definidas anteriormente, como esta explicado en el apartado 4.1

La function “**function _hook_get_roles_by_name()**”, explicada en el apartado 4.1, que agrega los roles propios del módulo en los roles definidos por defecto en Drupal

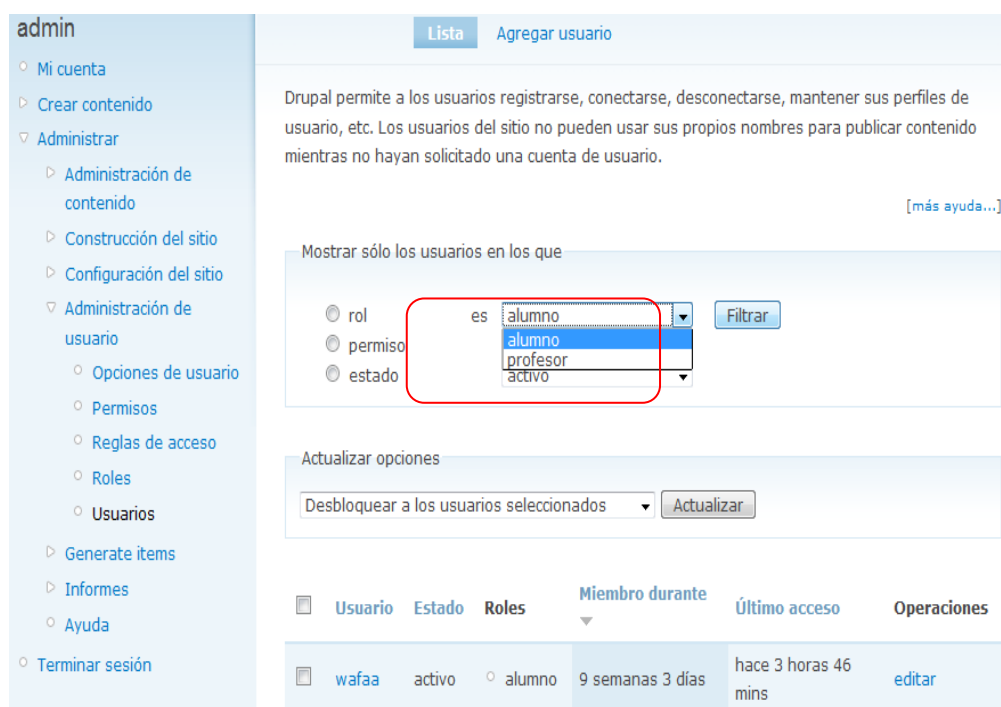


Figura 29 : Pantalla de usuarios

En la función “*desinstall*” se borran los roles instalados con la función “*install*” cuando se desinstala el módulo.

5 Instalación

Para la puesta en marcha de este proyecto es importante indicar los elementos imprescindibles para su desarrollo, la instalación de los elementos se puede realizar en cualquier plataforma, en el caso de este PFC se ha realizado en Windows:

- Ejecutable de “*Xampp 3.7*” que contiene SQL, PHP y Apache.
- Ejecutable de “*Drupal 6.20*”.
- Módulo de traducción “*es*” para introducir el español como idioma en la aplicación.
- Módulo del proyecto “*deliver-and-test*”.

La instalación completa de la aplicación se explica en detalle a continuación.

5.1 Instalación de Xampp

XAMPP es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. De hecho su nombre viene de ahí, X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP) y P (Perl). XAMPP es

independiente de la plataforma y tiene licencia GNU GPL. Existen versiones para Linux (testado para SuSE, RedHat, Mandrake y Debian), Windows (Windows 98, NT, 2000, XP y Vista), MacOS X y Solaris (desarrollada y probada con Solaris 8, probada con Solaris 9).

Para instalar el XAMPP se descarga el instalador XAMPP versión 1.7.3 en páginas webs como por ejemplo http://programa.com/descargar_xampp y se ejecuta siguiendo los pasos que se explican en las páginas de instalación XAMPP como por ejemplos <http://blog.suenyos.com/2010/07/16/instalar-xampp-windows-xp-vista-7/>.

5.2 Instalación de Drupal

Para el desarrollo de la aplicación se ha instalado la versión 6.20 de *Drupal* del sitio web <http://drupal.org>, hay versiones posteriores que pueden ser compatibles pero no se garantiza el correcto funcionamiento del módulo desarrollado porque no se han realizado pruebas al respecto.[27]

The screenshot shows the Drupal.org website. The 'Download & Extend' section is active, with the 'Distributions' tab highlighted by a red dashed circle. Below the tabs, there is a 'View' button and a 'Version control' link. The main content area contains text about downloading the official Drupal core files, a 'Before installing Drupal' section with links to system requirements and installation guide, and a 'Following Drupal core development' section with links to RSS feeds and Twitter. A table of recommended releases is shown at the bottom left, and a sidebar on the right lists maintainers and issues for Drupal core.

Version	Downloads	Date	Links
7.12	tar.gz (2.95 MB) zip (3.36 MB)	2012-Feb-01	Notes
6.25	tar.gz (1.05 MB) zip (1.22 MB)	2012-Feb-29	Notes

Figura 30: Pantalla de descargar Drupal 6.20

1. Se descarga el archivo tar.gz en cualquier directorio.

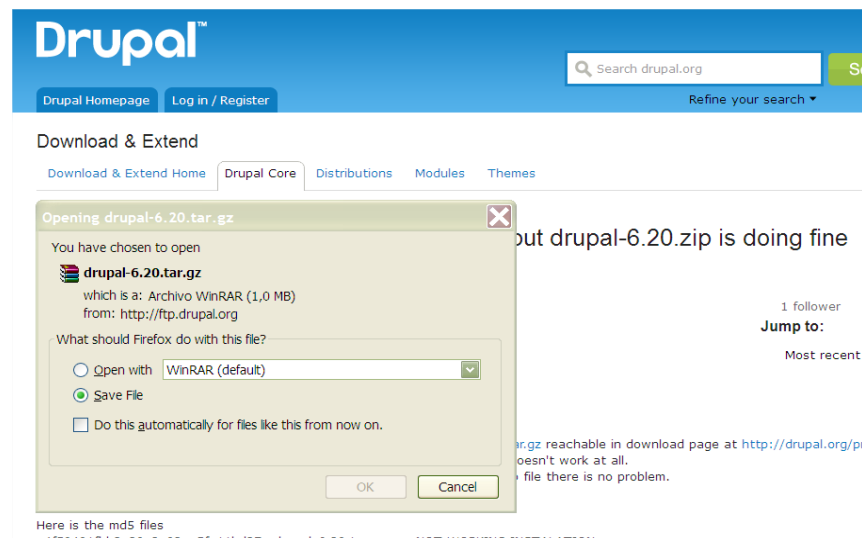


Figura 31: Pantalla de descargar Drupal 6.20

2. Se descomprime el archivo descargado en el directorio de su servidor web donde se encuentra el documento raíz del directorio HTML público. En nuestro caso se moverá a la carpeta Xampp/htdocs.

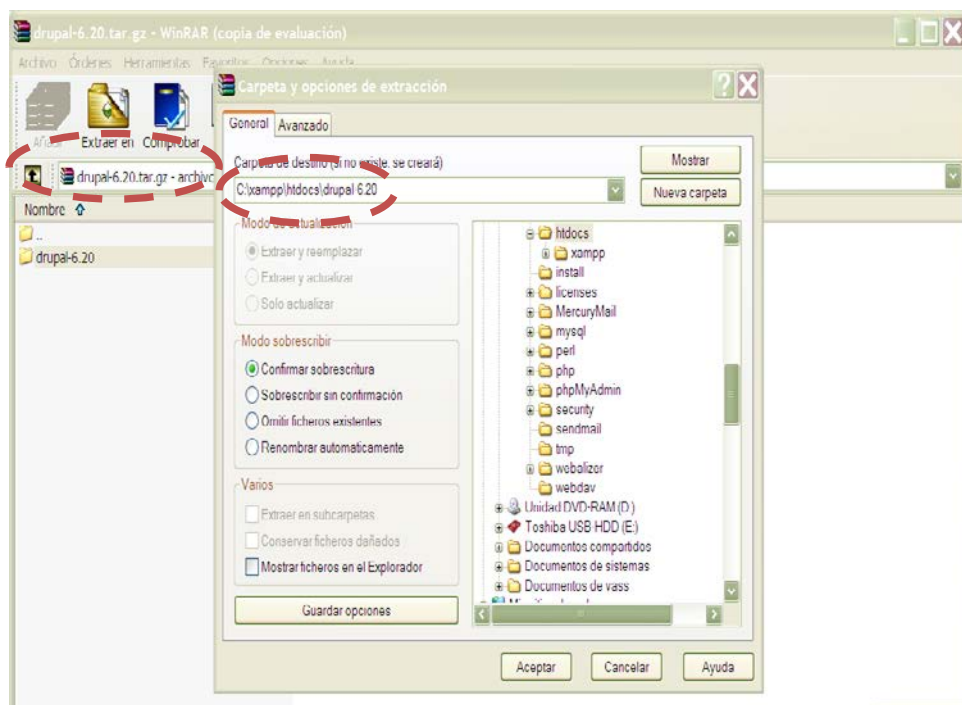


Figura 32: Pantalla de directorios de descarga

3. Se le cambia el nombre a la carpeta por otro nombre representativo para usarlo en la url de acceso a la aplicación. En el caso de este proyecto se ha nombrado la carpeta ***Deliver-and-test***.

5.3 Instalación de idioma

Para poder configurar Drupal en español hay que instalar un módulo que permite agregar el idioma en la configuración. Los siguientes pasos explican en detalle cómo conseguirlo:

1. Descargar el modulo "*es-6.x-1.4.tar*".

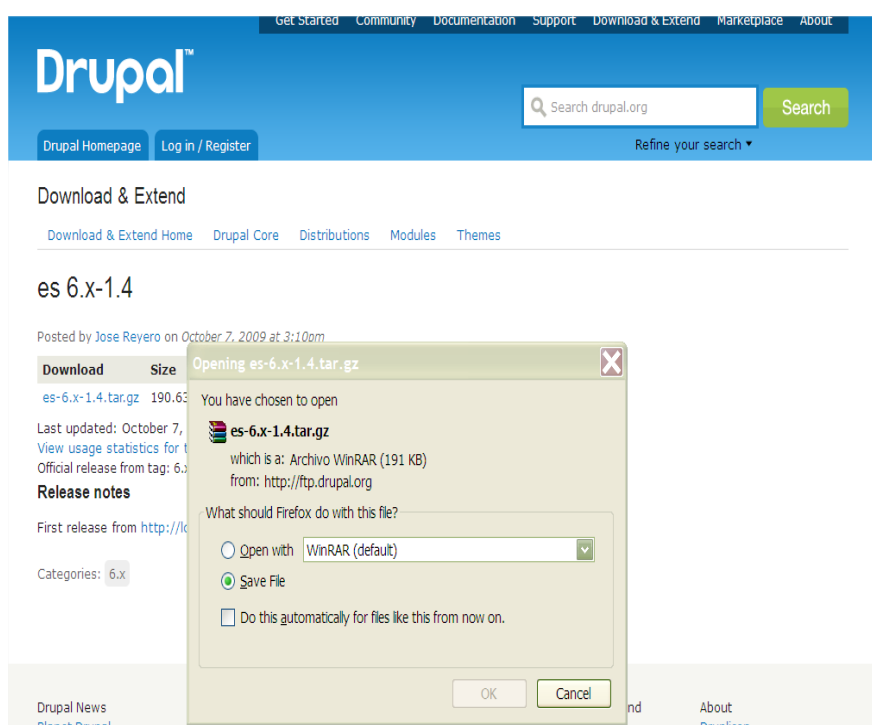


Figura 33: Pantalla de descargar la traducción español

2. Descomprimirlo en cualquier directorio y copiar los archivos de la carpeta "es-6.x-1.4" dentro de la carpeta: "drupal-**6.20**", instalada anteriormente, sobrescribiendo los archivos que sean necesarios o descomprimirlo directamente en el carpeta.

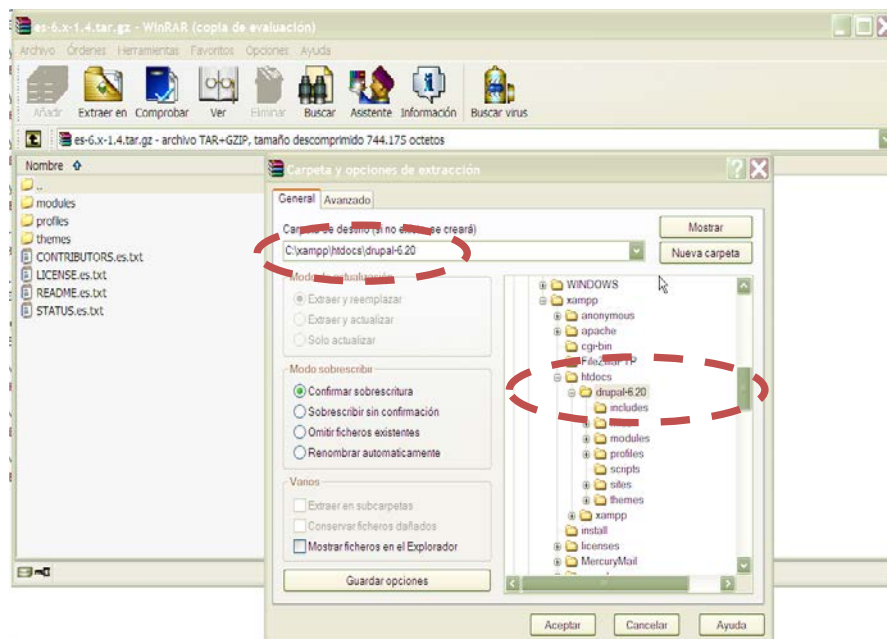


Figura 34: pantalla de directorio de descarga

5.4 Configuración de Drupal

Después de realizar todos los apartados anteriores, habrá que configurar Drupal para poder empezar a desarrollar. Para ello hay que seguir los pasos siguientes:

1. Abrir el panel de control de XAMPP desde Windows->todos los programas



Figura 35: pantalla de acceso a panel de control XAMPP

2. Arrancar Apache y SQL desde el panel de control abierto anteriormente

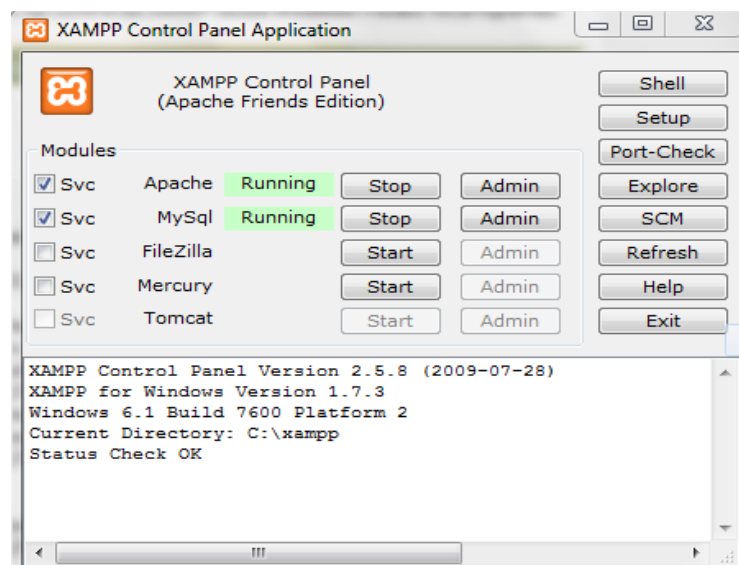


Figura 36: pantalla de panel de control

3. En el panel hay una opción de **"Admin"**, que sirve para configurar la base de datos que se utilizará en el proyecto.
 - a. Se pulsa en el botón **"Admin"** y se abre el sitio web **adminPHP**

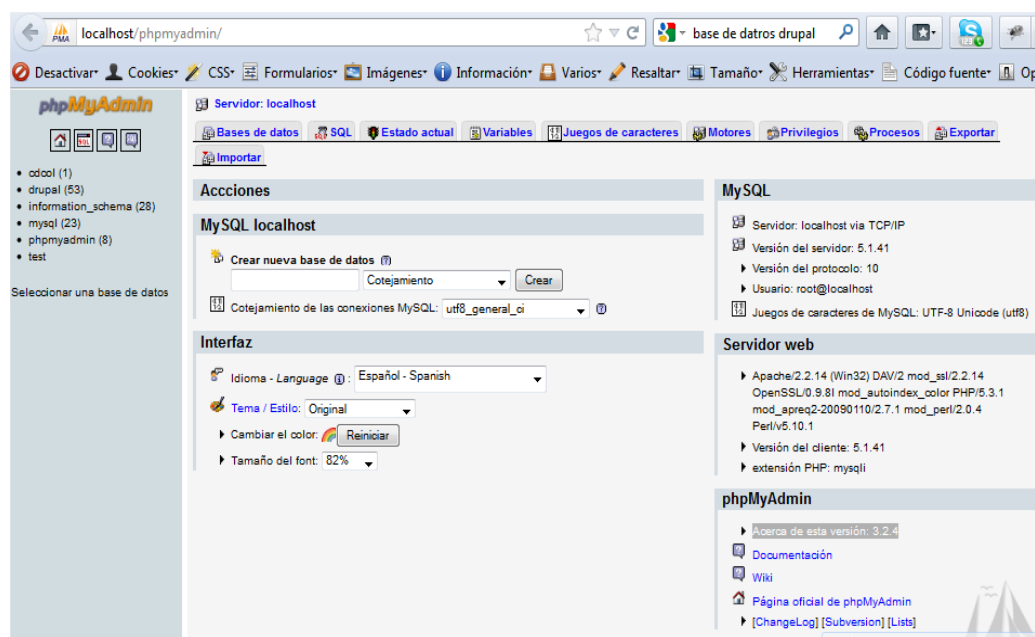


Figura 37: Pantalla de phpAdmin

- b. Se crea la base de datos que se utilizará en el proyecto. En este proyecto la llamamos drupal.

- c. Se crea el usuario administrador con los permisos correspondientes.

phpMyAdmin

Servidor: localhost

Bases de datos SQL Estado actual Variables Juegos de caracteres Motores Privilegios Procesos Exportar Importar

Agregar un nuevo usuario

Información de la cuenta:

Nombre de usuario: Use el campo de texto

Servidor: Cualquier servidor

Contraseña: Use el campo de texto

Debe volver a escribir:

Generar la contraseña: Generar

Base de datos para el usuario:

☒ Ninguna

☐ Crear base de datos con el mismo nombre y otorgue todos los privilegios

☐ Otorgue todos los privilegios al nombre que contiene comodín (username_%)

Privilegios globales (Marcar todos/as / Desmarcar todos)

Nota: Los nombres de los privilegios de MySQL están expresados en inglés

Datos	Estructura	Administración	Límites de recursos
<input type="checkbox"/> SELECT	<input type="checkbox"/> CREATE	<input type="checkbox"/> GRANT	MAX QUERIES PER HOUR 0
<input type="checkbox"/> INSERT	<input type="checkbox"/> ALTER	<input type="checkbox"/> SUPER	MAX UPDATES PER HOUR 0
<input type="checkbox"/> UPDATE	<input type="checkbox"/> INDEX	<input type="checkbox"/> PROCESS	MAX CONNECTIONS PER HOUR 0
<input type="checkbox"/> DELETE	<input type="checkbox"/> DROP	<input type="checkbox"/> RELOAD	MAX USER CONNECTIONS 0
<input type="checkbox"/> FILE	<input type="checkbox"/> CREATE TEMPORARY TABLES	<input type="checkbox"/> SHUTDOWN	
	<input type="checkbox"/> SHOW VIEW	<input type="checkbox"/> SHOW DATABASES	
	<input type="checkbox"/> CREATE ROUTINE	<input type="checkbox"/> LOCK TABLES	
	<input type="checkbox"/> ALTER ROUTINE	<input type="checkbox"/> REFERENCES	
	<input type="checkbox"/> EXECUTE	<input type="checkbox"/> REPLICATION CLIENT	
	<input type="checkbox"/> CREATE VIEW	<input type="checkbox"/> REPLICATION SLAVE	
	<input type="checkbox"/> EVENT	<input type="checkbox"/> CREATE USER	
	<input type="checkbox"/> TRIGGER		

Continuar

Figura 38: Pantalla de crear un usuario administrador

Se ha creado una base de datos y los datos del administrador que la gestionará.

4. Abrir un navegador y acceder a nuestro sitio Drupal con la siguiente url `http://Localhost/[Nombre de la carpeta]` en el caso de este proyecto será `http://Localhost/PFC`.
5. Se Crea una cuenta siguiendo los pasos de instalación indicados en las páginas web [27]

En los datos de la base de datos se insertan los datos creados en el paso 3, nombre de la base de datos, administrador.

5.5 Instalar el módulo “Deliver-and-test”

Para instalar el módulo del proyecto hay que realizar los siguientes pasos:

1. Copiar la carpeta del módulo “Deliver-and-test” en el directorio C:\xampp\htdocs\PFC\modules

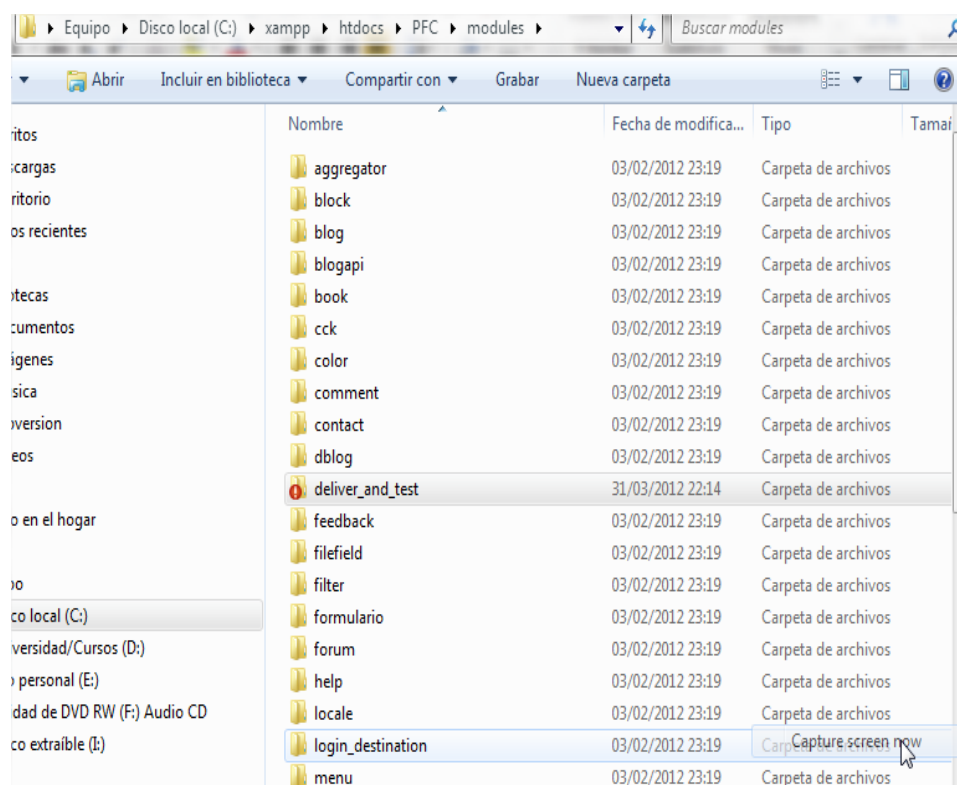


Figura 39: Directorio del module

2. Acceder desde la URL de Drupal <http://Localhost/PFC>.



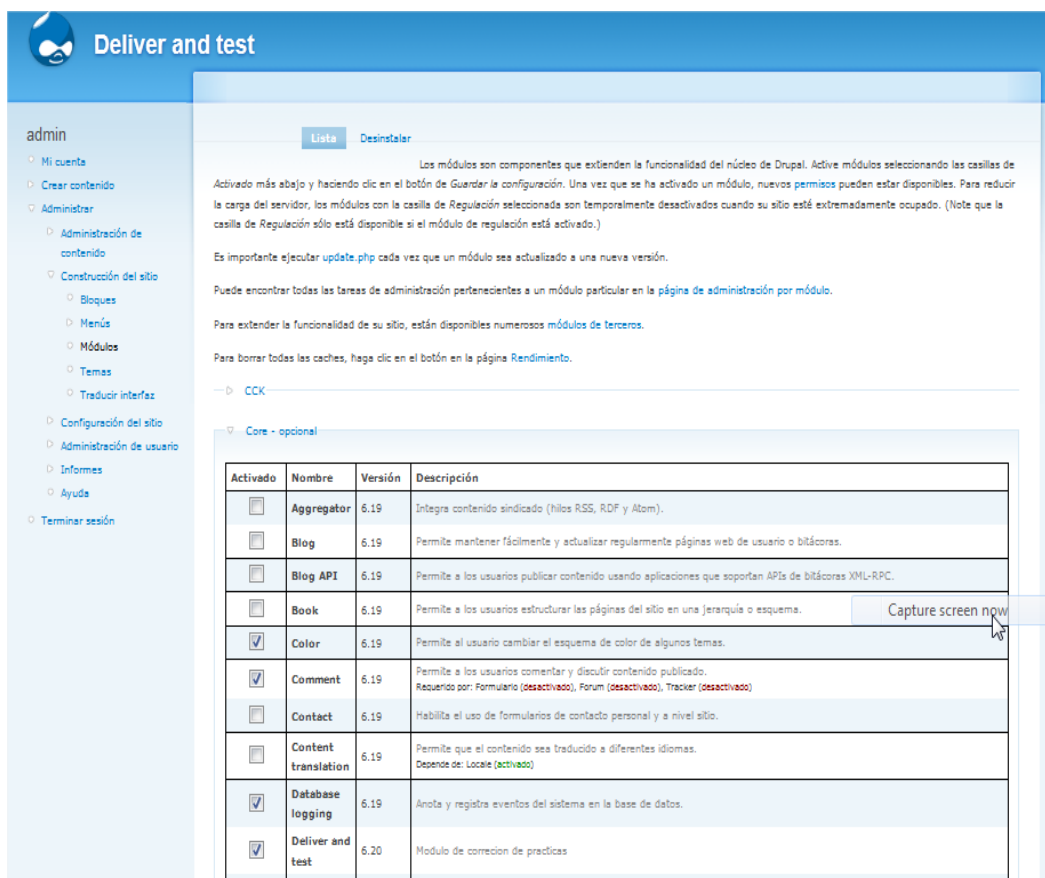
Figura 40: pantalla de acceso a la aplicación web “deliver-and-test”

3. Autenticarse como administrador, con los datos creados en la base de datos.



Figura 41: pantalla de autenticación

4. Ir a administrador→módulos y agregar el módulo



Deliver and test

admin

- Mi cuenta
- Crear contenido
- ▽ Administrar
 - ▷ Administración de contenido
 - ▷ Construcción del sitio
 - Bloques
 - Menús
 - Módulos
 - Temas
 - Traducir interfaz
 - ▷ Configuración del sitio
 - ▷ Administración de usuario
 - ▷ Informes
 - ▷ Ayuda
- Terminar sesión

Lista Desinstalar

Los módulos son componentes que extienden la funcionalidad del núcleo de Drupal. Active módulos seleccionando las casillas de Activado más abajo y haciendo clic en el botón de Guardar la configuración. Una vez que se ha activado un módulo, nuevos permisos pueden estar disponibles. Para reducir la carga del servidor, los módulos con la casilla de Regulación seleccionada son temporalmente desactivados cuando su sitio esté extremadamente ocupado. (Note que la casilla de Regulación sólo está disponible si el módulo de regulación está activado.)

Es importante ejecutar `update.php` cada vez que un módulo sea actualizado a una nueva versión.

Puede encontrar todas las tareas de administración pertenecientes a un módulo particular en la [página de administración por módulo](#).

Para extender la funcionalidad de su sitio, están disponibles numerosos [módulos de terceros](#).

Para borrar todas las caches, haga clic en el botón en la página [Rendimiento](#).

—> CCK

Core - opcional

Activado	Nombre	Versión	Descripción
<input type="checkbox"/>	Aggregator	6.19	Integra contenido sindicado (hilos RSS, RDF y Atom).
<input type="checkbox"/>	Blog	6.19	Permite mantener fácilmente y actualizar regularmente páginas web de usuario o bitácoras.
<input type="checkbox"/>	Blog API	6.19	Permite a los usuarios publicar contenido usando aplicaciones que soportan APIs de bitácoras XML-RPC.
<input type="checkbox"/>	Book	6.19	Permite a los usuarios estructurar las páginas del sitio en una jerarquía o esquema.
<input checked="" type="checkbox"/>	Color	6.19	Permite al usuario cambiar el esquema de color de algunos temas.
<input checked="" type="checkbox"/>	Comment	6.19	Permite a los usuarios comentar y discutir contenido publicado. Requiere por: Formulario (desactivado), Forum (desactivado), Tracker (desactivado)
<input type="checkbox"/>	Contact	6.19	Habilita el uso de formularios de contacto personal y a nivel sitio.
<input type="checkbox"/>	Content translation	6.19	Permite que el contenido sea traducido a diferentes idiomas. Depende de: Locale (activado)
<input checked="" type="checkbox"/>	Database logging	6.19	Anota y registra eventos del sistema en la base de datos.
<input checked="" type="checkbox"/>	Deliver and test	6.20	Módulo de corrección de prácticas

Capture screen now

Figura 42: pantalla de activar módulo

6 Conclusión y trabajos futuros

6.1 Conclusión

En este proyecto fin de carrera se han cumplido todos los objetivos que se han marcado al principio desarrollando una aplicación que ofrece a los profesores y alumnos las funcionalidades establecidas para permitir la gestión, entrega y corrección de prácticas on-line.

Antes de todo se han analizado los requisitos funcionales de cada uno de los roles que se han definido para este proyecto. Para el profesor se han definido las funciones más importantes para mejorar su gestión y corrección de las prácticas como facilitarle la subida de sus test, enunciados e interfaces, permitirle el acceso a de forma automática a las practicas que han superado un test, tener un seguimiento de los esfuerzos de sus alumnos, puntuar de forma automática a los alumnos, añadir comentarios sobre una práctica, modificar un test y que se notifique automáticamente a los alumnos que han elegido el test modificado para pasar sus prácticas. Para el alumno, se han definido funciones que le faciliten el desarrollo de su práctica, la entrega y la comunicación con el profesor como subir tantas versiones de la práctica como quisiera, pasar el test que quiere, acceder al fichero de un test, enunciado e interfaz, visualizar los errores de compilación y de ejecución de su práctica, mejorar su metodología en programación, visualizar los comentarios y la nota puestas por el profesor y evitar errores de entrega.

Después de analizar los requisitos funcionales para la aplicación web, se ha procedido a la elección de la plataforma a utilizar para desarrollar la aplicación "Deliver-and-test" y se ha decidido usar un gestor documental para satisfacer las necesidades de la aplicación y dentro de los 4 gestores documentales más usados, **Alfresco**, **Plone**, **Drupal** y **Joomla** se ha elegido Drupal. Se plantean dos formas de desarrollar, la primera es usando sus propias funciones definidas y creadas por otros usuarios y la segunda es crear un nuevo módulo independiente, por motivo de independencia se ha optado por la segunda opción que se ha considerado la mejor. Teniendo claro la plataforma y el tipo de desarrollo que se va a hacer se ha procedido a estudiar los casos de uso de los dos roles y la implementación de cada función que se ha definido en los requisitos funcionales.

La realización de la aplicación "**Deliver-and-test**" me ha brindado la oportunidad de adquirir conocimientos en la tecnología Drupal y en el lenguaje de programación PHP, que hasta el primer día de análisis eran desconocidos. Aunque su aprendizaje es lo que más me ha costado y que ha sido una de las tareas que más tiempo se le ha dedicado, vale la pena el esfuerzo por ser una plataforma de futuro que se utiliza para proyectos potentes, fácil

de mantener y cualquier usuario con conocimiento PHP puede mejorar el módulo ampliando sus funcionalidades.

6.2 Trabajos futuros

El proyecto fin de carrera se hizo como módulo para soportar un crecimiento de las funcionalidades y aceptar mejoras. En este proyecto solo se han satisfecho las necesidades básicas y necesarias para facilitar la gestión y corrección de prácticas pero se pueden agregar muchas mejoras y ampliar las funcionalidades. Algunas de las mejoras que se pueden realizar son:

1. Mensajes

Un mensaje puede ser muy útil tanto para el alumno como el profesor para transmitir dudas o notificaciones. El alumno puede tener dudas sobre el test que quiere superar y un mensaje enviado desde la aplicación web puede ser un medio fácil de transmitir esas dudas al profesor sin tener que asistir a tutorías o enviarlo desde otra aplicación. El profesor pueda contestar a los mensajes recibidos por los alumnos mediante mensajes también desde la aplicación “deliver-and-test”.

Se puede agregar en la pantalla principal del profesor y alumno un botón “mensaje” que permite acceder a un formulario de contacto y enviar los datos insertados al destinatario. Para ello hace falta crear una función nueva que crea un nuevo formulario y un botón “enviar” que envía los datos insertados al destinatario, los datos del destinatario se pueden obtener desde la base de datos.

2. Diversidad de lenguaje de programación

La aplicación actual está enfocada hacia el lenguaje de programación Java, se compila y se ejecuta en **Java**. Como mejora se puede cambiar el código implementado para que la aplicación pueda admitir cualquier lenguaje de programación y para ello hay que modificar la función **compilaTestP** que existe en las clases **deliver_and_test.profesor.inc** y **deliver_and_test.alumno.inc** cambiando los comandos de compilación y ejecución para que se elijan los comandos adecuados según el lenguaje de programación de la práctica. Hay que quitar también todas las comprobaciones del tipo de documento que se hacen en los formularios de “anadirTest”, “modificarTest”, “aniadirPractica” y “aniadirVersion”.

3. Usabilidad

El formato del texto devuelto como resultado de la compilación o la ejecución de la práctica es legible pero se puede mejorar dándole un formato más elegante. Se puede mejorar también el diseño de la aplicación realizando algunos cambios en la interfaz de usuario para que sea más agradable y no solo práctica.

4. Cambiar de rol en la misma sesión

Una persona que tenga usuarios con dos roles, pueda cambiar de usuario en su propia sesión, es decir, que pueda autenticarse como profesor y en la misma sesión pueda cambiar a alumno y viceversa. Una forma de implementarlo es poner un botón en la página principal que permite, al pulsarlo, acceder a la página principal del otro rol.

7 PRESUPUESTO

7.1 Costes personales

7.1.1 Investigación

Se ha llevado a cabo una investigación sobre las herramientas que pueden cubrir las necesidades de la aplicación web para elegir la más adecuada. Una vez elegida la herramienta se ha procedido a familiarizarse con ella y adquirir el conocimiento necesario para poder utilizarla y desarrollar el proyecto.

El tiempo invertido en aprendizaje e instalación de la herramienta: 6 semanas

7.1.2 Toma de requisitos

Se ha logrado identificar el objetivo del proyecto y determinar las tareas para implementar el proyecto, en 4 semanas.

7.1.3 Diseño del sistema

Después de analizar los requisitos del proyecto se ha procedido al diseño del sistema:

- Diseño de las tareas que realiza la aplicación
- Diseño de la estructura de la base de datos

El diseño ha durado 2 semanas

7.1.4 Desarrollo de la aplicación web

El desarrollo se ha hecho en PHP siguiendo las reglas y estructura de Drupal. La implementación ha durado 20 semanas

7.1.5 Pruebas

Las pruebas de la aplicación implementada tenían como objetivo comprobar el correcto funcionamiento de la aplicación y el cumplimiento de los requisitos tomados a principio del proyecto. Para ello se ha instalado Drupal, PHP y SQL en **localhost** y se han probado todos los casos de uso planteados en el punto [30].

La duración de las pruebas es de 2 semanas.

7.1.6 Instalación de drupal

Antes de instalar, se ha documentado sobre los pasos de instalación. En este proyecto había dos posibles formas de instalar Drupal, PHP y SQL. La primera es instala los 3 componentes por separado, descargar desde su la página oficial de cada producto la última version e instalarlo después realiza la conexión entre la base de datos y drupal.

La segunda forma es instalar un servidor que engloba los productos anteriores y arrancarlos desde su panel de control. En este proyecto se ha optado por la segunda opción y se ha tardado 1 día.

7.1.7 Desarrollo de la memoria

La última fase del proyecto es la elaboración de la memoria donde engloba la descripción de todas las fases del proyecto.

La duración de la memoria es de 9 semanas.

Tareas	Duración(en semanas)
Investigación	6
Toma de requisitos	4
Diseño del sistema	2
Desarrollo de la aplicación web	20
Pruebas	2
Instalación Drupal	1/7
Desarrollo de la memoria	9

Figura 43: Tabla con la duración de las tareas en semanas

7.2 Coste total

7.2.1 Costes materiales

Para la realización del proyecto se ha requerido:

- Un ordenador portátil 650 euros
- Un servidor central que puede costar 750 euros.

7.2.2 Costes de recursos humanos

Considerando unos honorarios de 40 euros/h para un ingeniero Técnico, precio actual de una consultora, el coste total del desarrollo de la aplicación es de 68.960 euros.

COSTES	CANTIDAD(euros)
Recursos humanos	68.960 €
Recursos materiales	1400 €

Figura 44: Resumen de los gastos del desarrollo del proyecto

8 Bibliografía

Páginas web

- [1] David M.Arnow & Oleg Barshay. On-line programming examinations using web to teach.
- [2] David Jakson & Michelle Usher. Grading student programs using ASSYST
- [3] Edward L.Jones. Grading student programs-A software testing
- [4] Jaime Spaccoo, David Hovemeyer, William Pugh, Fawzi Emad, Jeffrey K.Hollingsworth and Nelson

Padua-Perez. Experiences with Marmoset: Designing and using an advanced submission and testing system for programming courses.
- [5] Edward L. Jones. Grading student programs-A software testing approach
http://www.cis.famu.edu/~testlab/Papers/test_approach_to_grading.pdf
- [6] Artículo *Planetacms* de CMS
<http://www.planetacms.com/>
- [7] David M.Arnow. When you grade that: using e-mail and the network in programming course
- [8] Artículo Wikipedia de Sistema de gestión de contenidos
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos
- [9] Página oficial de Drupal
<http://www.drupal.org>
- [10] Página oficial de Joomla
<http://www.joomla.org/>
- [11] Página oficial de Zope
<http://www.zope.org>
- [12] Página oficial de Python
<http://www.python.org>
- [13] Página oficial de Alfresco
<http://www.alfresco.com/es/>
- [14] Página oficial de lenya apache
<http://lenya.apache.org/>
- [15] Media Wiki
<http://www.mediawiki.org/wiki/MediaWiki/es>

- [16] Página oficial de PHPBB
<http://www.phpbb.com/>

- [17] Página oficial de MYBB
<http://www.mybb.com/>

- [18] Características de Drupal
<http://drupal.org.es/caracteristicas>
Cursos básicos de Drupal
<http://www.educasitio.com/curso-basico-drupal>

Tutorial Drupal
<http://www.pilos.com.co/drupal/18-video-tutoriales-de-drupal-basico/>
<http://drupal.org/getting-started/before/overview>

- [19] Página oficial de Plone
<http://plone.org/>

- [20] Fátima Sanchez. PFC "Análisis, diseño e implementación de una aplicación web para gestión de información policial municipal basada en Plone".

- [21] página oficial Drupal, Api
<http://api.drupal.org/api/drupal/includes!module.inc/group/hooks/6>

Api Drupal
<http://api.drupalecommerce.org/api/drupal/drupal--includes--module.inc/group/hooks/6.x>

- [22] Artículo Wikipedia PHP
<http://es.wikipedia.org/wiki/PHP>

- [23] Tutorial PHP
<http://www.php.net/manual/en>

- [24] Página oficial MYSQL
<http://www.mysql.com/why-mysql/>

- [25] Artículo Wikipedia Apache
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

- [26] Página oficial Apache Friends- XAMMP
<http://www.apachefriends.org/es/xampp.html>

- [27] Manual de instalación Drupal
<http://drupal.org.es/manuales/instalacion>

ANEXO

Los tres anexos que representan el diagrama de flujo de alumno, docente y base de datos están en el CD adjunto porque no se ha podido agregarlos en la memoria debido al gran tamaño de los ficheros. Los anexos están hechos con el programa Microsoft Office Visual (VSD). A continuación se detalla el contenido de esos anexos:

Anexo I

Representa el flujo completo de las funcionalidades del docente siguiendo los requisitos funcionales mencionados en la sección (2.1.1).

Anexo II

Representa el flujo completo de las funcionalidades del docente siguiendo los requisitos funcionales mencionados en la sección (2.1.2).

Anexo III

Representa la distribución de las tablas, propias de la aplicación, y sus contenidos.